Geometric, Combinatorial, and Experimental Perspectives on the k-Max-Cut Problem

by

Teresa L. Chambers

B.Sc., University of Pittsburgh; Pittsburgh, PA, 2019

B.Eng., University of Pittsburgh; Pittsburgh, PA, 2019

M.Sc., Brown University; Providence, RI, 2021

A dissertation submitted in partial fulfillment of the

requirements for the degree of Doctor of Philosophy

in The Division of Applied Mathematics at Brown University

PROVIDENCE, RHODE ISLAND

May 2025

This dissertation by Teressa L. Chambers is accepted in its present form
by The Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date________________ ______________________________________

Caroline Klivans, Ph.D., Advisor

Recommended to the Graduate Council

Date________________ ______________________________________

Melody Chan, Ph.D., Reader

Date________________ ______________________________________

Alice Paul, Ph.D., Reader

Approved by the Graduate Council

Date________________ ______________________________________

Peter M. Weber, Dean of the Graduate School

Vita

Education

Brown University *August 2019 – Expected May 2025*

Ph.D., Applied Mathematics, Expected May 2025

Masters, Applied Mathematics, May 2021

University of Pittsburgh *August 2014 – May 2019*

B.Eng., Bioengineering

B.Sc., Mathematics

Publications

M. Bayer, S. Borgwardt, T. Chambers, S. Daugherty, A. Dawkins, D. Deligeorgaki, H.-C. Liao, T. McAllister, A. Morrison, G. Nelson, A. Vindas-Melendez. *Combinatorics of Generalized Parking-Function Polytopes.* Preprint at https://arxiv.org/abs/2403.07387

Teaching

Brown University

Instructor, Statistical Inference I *Summer 2024*

Head Teaching Assistant, Essential Statistics *Spring 2023*

Head Teaching Assistant, Operations Research: Deterministic Models *Fall 2023*

Teaching Assistant, Recent Applications of Probability and Statistics *Spring 2021*

Teaching Assistant, Operations Research: Deterministic Models *Fall 2020*

University of Pittsburgh

Teaching Assistant, Biotransport Phenomena	*Fall 2018, Spring 2019*
Teaching Assistant, The Art of Making	*Fall 2015, Spring 2016*

Outreach and Service

Directed Reading Program Mentor	*Fall 2021 through Spring 2025*
APMA DEI Committee Graduate Representative	*Fall 2021 through Spring 2024*
Graduate Student Council, various roles	*Fall 2021 through Spring 2025*
Graduate Library Advisory Committee Member	*Fall 2020 through Fall 2024*
Math Resource Center Tutor	*Fall 2020, Spring 2021*

Graduate Coursework

Real Analysis

Functional Analysis

Numerical Solutions to Partial Differential Equations I and II

Theory of Probability I

Combinatorial Theory

Nonlinear Dynamical Systems I and II

Algebra I and II

Preface and Acknowledgments

Thank you first and foremost to my advisor, Caroline Klivans, for six years of steadfast guidance and support. I knew I wanted to work with her from the day I interviewed at Brown, and I have been bragging about her incredible skill as a mentor and advisor ever since.

Thank you also to Melody Chan and Alice Paul for making time in a very hectic semester to read this dissertation and provide feedback and insight. In the field of mathematics, I am delighted to have had the opportunity to assemble an all-female thesis committee that so perfectly fits my research topics.

Thank you to my cohort, who have walked this path with me from the start of our first year, through the COVID-19 pandemic, navigating prelim exams and mask mandates, doing our best to maintain extracurricular opportunities and organizations in the department in order to pass them on to future students. Research in our time here has been a difficult and sometimes lonely endeavor, with fewer opportunities for in-person collaborations and not much overlap among our respective projects, but we all pushed through to find our own ways. Thank you especially to Erik, Kevin, Juniper, and Tim, for joining me in building supportive and engaging office environments filled with jokes and nonsense and deep conversations. We made it.

Thank you to the Caroline Klivans combinatorics research group at Brown, which has finally taken shape across the Math and Applied Math programs, for validating my passion for discrete math. Nick, Yifan, Ethan, Andrew, Carlos, and Sarah have given me community and camaraderie as each has joined Carly's group, and it has been a true delight to discuss math with each of them.

I would not be here without the love and support of my family and friends over all these years. Thank you to my parents, for always being convinced that math is where I belong; my sister Sarah, for consistently making me laugh at just the right moment; my brother Mark, for reminding me why this journey is so important to me; and my sister Katy Anne, for steady wisdom that sometimes makes me question which of us is older. Thank you also to Megan, Chris, Josh, and Nick, who have supplied me with a monthly dose of magic and mayhem for six incredible years, as well as plenty of individual patience, reassurance, and kindness throughout this process.

Finally, thank you to my wonderful husband Erik Bergland, whom I met on the first day of orientation and fell for in less than a month. He made a beautiful pun for me when dedicating his thesis, and I simply cannot come up with a better one, so I'll quote him: "That we met during this time in our lives truly is providence."

Contents

List of Figures

Chapter One

Introduction

The dissertation is divided into four large parts. The first part, "On Partition Polytopes," covers the main class of polytopes studied by the author, which are linked to an interpretation of the k-max-cut problem as a partitioning problem. This details both modeling observations and theoretical results regarding these types of polytopes. The second part, "On the Max-Cut Problem," details a theoretical and computational analysis of a recent algorithm for solving the k-max-cut problem using semidefinite programming. The method was assessed broadly, operating over a specific collection of convex bodies. To further assess its potential as a clustering algorithm, the method was implemented and analyzed in a variety of experimental clustering settings. The third and fourth parts, "On Parking-Function Polytopes" and "On Arboricity," cover two areas of study for the author that are further afield from the main thesis work. One is a joint project on the polytopes for $\mathbf{b}$-parking functions, characterizing their structure, behavior, and relationships to other known polytope classes. The other focuses on a graph invariant known as the arboricity polynomial, developing tools to break down the polynomials according to graph structure. Each of these parts consists of a single larger chapter detailing the results.

1.1 Summary of Partition Polytope Results

The first part of the dissertation centers on partition polytopes: polytopes whose vertices represent partitions of $[n]$ into at most k blocks. Any solution to the k-max-cut problem is a partition, and these polytopes act as relaxed solution spaces for the problem. Most prior results pertaining to these polytopes focus on their facets, or restrict to the case $k = 2$ ([6], [19], [22], [23], [24], [25], [34]). Our work specifically examines the skeletons and other lower-dimensional structures of the polytopes for larger values of k, none of which have previously been characterized.

Chapter 3 provides an examination of small ($n = 4, 5$) partition polytopes through models constructed in Sage [64]. The f-vectors of each polytope are computed for reference. Vertex locations and facet-defining hyperplanes are explicitly provided for the $n = 4$ polytopes. The edge structures are fully elucidated for every $n = 4$ and $n = 5$ polytope, and trends in the structures observed across multiple polytopes are concretized to generate theoretical questions. Additionally, some low-dimensional faces are identified that possess structural similarities to the full-scale polytopes.

In Chapter 4, several of the case study observations are proved for general n, k. Full neighborhoods are characterized for the unique single-block partition of $[n]$ and the unique n-block partition of $[n]$, which exhibit special behavior in the polytopes (Proposition 4.1.1, Proposition 4.1.3). A subset of edges that are similar to connections in the partition lattice are also proved to exist in general (Proposition 4.1.8). Based on observations that the polytope skeleton possesses many more edges than the lattice, edges between vertices representing partitions with $n - 1$ blocks are fully characterized for both the $k = n - 1$ and $k = n$ polytopes (Proposition 4.1.9, Proposition 4.1.10). A variety of additional conjectures are put forth for future inquiry. In the main result of this section, the low-dimensional faces with structural similarities to smaller partition polytopes are proven to exist in generality, nested within each other and combinatorially equivalent to partition polytopes themselves (Lemma 4.2.1, Theorem 4.2.4, Corollary 4.2.5). This final result indicates that the family of polytopes is much more interconnected than previously known.

1.2 Summary of Max-Cut Results

The portion of the dissertation which centrally considers the k-max-cut problem revolves around a particular approach to relaxing and solving the standard combinatorial problem, which is NP-hard. The approach in question is a semidefinite programming relaxation pioneered by Goemans and Williamson [32], accompanied by a rounding algorithm developed by Felzenszwalb, Klivans, and Paul to recover combinatorial solutions from the semidefinite construction [29]. The solution space for this relaxation is a curious object with both polytopal and non-polytopal boundary characteristics, called the elliptope.

Chapter 6 evaluates the rounding algorithm and its properties independent of the max-cut setting, divided into consideration of polytopal and non-polytopal convex boundaries. Non-polytopal boundaries are examined via a case study into behavior over the two-dimensional ellipse, examining how the non-polytopal boundary properties and the locations of the ellipse in $\mathbb{R}^2$ influence the fixed points of the rounding algorithm over the boundary. Polytopal boundaries are examined more broadly, accompanied by an in-depth inquiry into saddle points. A particular subgroup of saddle points, called full-measure saddle points, is defined and characterized in the polytope setting (Lemma 6.2.6). It is also shown that these saddle points can be iteratively constructed in arbitrarily many dimensions by taking orthogonal prisms of polytopes (Proposition 6.2.9 and Corollary 6.2.10).

Chapter 7 focuses on the joint operation of the rounding algorithm with the semidefinite relaxation for which it was designed, in the context of data clustering.

Any data clustering problem can be viewed as a k-max-cut problem on the complete graph, making the clustering setting a promising use case for the full approximation algorithm. A variety of experiments were undertaken to assess the utility of semidefinite programming with the rounding algorithm for these problems. Tests were conducted using data generated from regularized two-dimensional Gaussian distributions and randomly selected from the MNIST data set, and the approximation's performance as an algorithm was assessed using three different clustering scores in comparison to an algorithm that is widely used in industry. In these tests, it was observed that the semidefinite programming approach yielded clear advantages in identifying the true number of clusters in a dataset, but exhibited a preference for evenly-balanced categorizations of data (Figure 7.3, Figure 7.4, and Figure 7.7). The observations recorded here yield several intriguing follow-up questions.

1.3 Summary of Parking-Function Polytopes

This part of the dissertation considers polytopes generated from a particular generalization of parking functions known as **b**-parking functions. These results were obtained in collaboration with Margaret Bayer, Steffen Borgwardt, Spencer Daugherty, Aleyah Dawkins, Danai Deligeorgaki, Hsin-Chieh Liao, Tyrrell McAllister, Angela Morrison, Garrett Nelson, and Andrés Vindas-Meléndez, and the chapter discussing the project primarily reproduces a paper appearing on arXiv in 2024 [8]. The facets, vertices, and edges of any **b**-parking function polytope are fully described (Theorem 8.2.3, Proposition 8.2.6), and the polytopes are shown to be generalized permutahedra (Theorem 8.2.11). By examining the polytopes through the lens of building-sets, we further prove that any **b**-parking function polytope is either a stellohedron or is equivalent to a polytope generated by classical parking functions (Corollary 8.3.14). Links are also constructed that envision the polytopes as polymatroids (Theorem 8.4.3), relaxed Birkhoff polytopes (Theorem 8.5.1), and relaxed bounded-size partition polytopes (Corollary 8.5.2).

1.4 Summary of Arboricity

The final part of the dissertation covers results pertaining to the arboricity polynomial, a graph invariant similar to the chromatic polynomial but with much less nice behavior. In particular, the arboricity polynomial does not have a closed formula for inductive construction based on the polynomials of smaller graphs, nor are there known formulae for these polynomials for any graph families other than trees and cycles. Here, new tools are developed to allow the polynomials to be broken down and understood in certain special cases, such as when a graph is 1-edge-connected (Proposition 9.3.1 and Proposition 9.3.2) and when a graph has a vertex of degree 2 (Proposition 9.3.3). In addition, we show that edge contraction and deletion yield polynomials that bound the arboricity polynomial for a given graph (Proposition 9.3.4). Alongside these results, we provide a bank of example arboricity polynomials to demonstrate variability within and across graph families. Since the arboricity polynomial in the standard basis can be intricate and difficult to comprehend, we also show that the polynomial has many nice properties and clean interpretations in the falling-factorial basis (Proposition 9.4.2).

Part I

On Partition Polytopes

Chapter Two

Background: Polytopes and Partitions

Many interesting families of polytopes are defined according to some collection of combinatorial objects. The category investigated in this part of the dissertation is up-to-k-partition polytopes, each of which is formed by representing a specific collection of partitions as vectors and taking their convex hull. Here, we explore the literature underlying these polytopes to provide a solid background before moving on to the results. We will establish some principles about polytopes and partitions independently of each other before combining them to inspect partition polytopes. One important contextual element for these polytopes is their connection to the k-max-cut problem, which will be made explicit in Section 2.3.

2.1 Key Concepts in Polytope Theory

We will in general assume a foundational understanding of polytopes, such as is found in Ziegler's *Lectures on Polytopes* [74]. However, some key terminology and results will be replicated here for convenience.

Definition 2.1.1. A **polytope** $P \subset \mathbb{R}^d$ is the convex hull of some finite set of points in $\mathbb{R}^d$, or equivalently the bounded intersection of finitely many closed half-spaces.

Polytopes under this pair of equivalent definitions are convex bodies of at most d dimensions. A polytope can be lower in dimension than the space in which it resides. The half-spaces can be represented as inequalities $\mathbf{cx} \leq c_0$, where $\mathbf{c}$ is a collection of d coefficients, $\mathbf{x} \in \mathbb{R}^d$ is any vector, and c_0 is a constant; any inequality over $\mathbb{R}^d$ such that all points in P satisfy the inequality is called *valid* for the polytope P.

Definition 2.1.2. A **face** F of some polytope P is a set of the form

$$F = P \cap \{\mathbf{x} \in \mathbb{R}^d : \mathbf{cx} = c_0\}$$

where $\mathbf{cx} \leq c_0$ is a valid inequality for the polytope P.

The boundary of P is comprised of its faces. Zero-dimensional faces are called *vertices*, one-dimensional faces are called *edges*, and for an n-dimensional polytope, $n \leq d$, the $(n-1)$-dimensional faces are called *facets*. P is itself a face, and any intersection of faces is also a face of P. Moreover, each face F is also a polytope, whose faces are exactly the faces of P it contains. Since polytopes will have faces of every dimension from 0 up to n, it is convenient to represent the number of faces of each dimension in one place.

Definition 2.1.3. The f**-vector** of a n-dimensional polytope P is a vector with $n+2$ elements

$$f_P = (f_{-1}, f_0, f_1, ..., f_{n-1}, f_n)$$

where f_i is the number of faces of P of dimension i.

In this description, $i = -1$ is the singular empty face $\varnothing$ and $i = n$ is the full polytope P, so $f_{-1} = f_n = 1$ for any polytope.

It is often useful to consider a few well-understood properties that a polytope might possess. Each of these properties describes a nice and well-understood regularity that can capture much of the internal structure of a polytope. We will use these to briefly summarize polytope models in Chapter 3.

Definition 2.1.4. A n-dimensional polytope is called **simple** if every vertex is contained in exactly n facets.

Definition 2.1.5. A n-dimensional polytope is called **simplicial** if every facet contains exactly n vertices.

Definition 2.1.6. A n-dimensional polytope is called k**-neighborly** if every subset of k or fewer vertices is the vertex set of some face.

Example 2.1.7. We can demonstrate each of the above concepts using a well-known three-dimensional polytope ($n = 3$). In three dimensions, the facets are two-dimensional faces. A standard cube (Figure 2.1, left) is a simple polytope - every vertex belongs to exactly 3 facets. An octahedron (Figure 2.1, middle) is a simplicial polytope - every facet is a triangle, containing exactly 3 vertices. A tetrahedron (Figure 2.1, right) is both simple and simplicial, and is also 4-neighborly. Every subset of 1 vertex is a 0-dimensional face, the vertex itself; every subset of 2 vertices is a 1-dimensional face, i.e. an edge; every subset of 3 vertices is a 2-dimensional face; and the full set of 4 vertices is a 3-dimensional face in the form of the full tetrahedron.

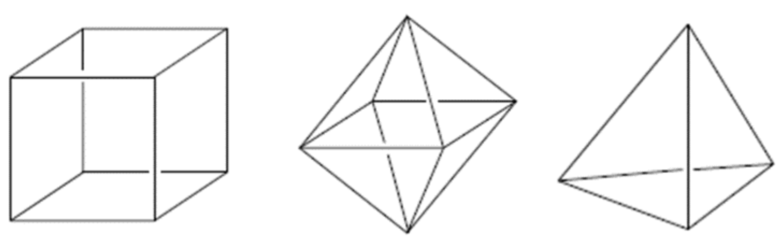


Figure 2.1: The three-dimensional cube, octahedron, and tetrahedron [74]

One important element for understanding polytopes and their structure is the *normal cone*, which is defined for any subset of a set in $\mathbb{R}^d$. For the purposes of this part of the thesis, the larger set will always be a polytope; however, the term is defined in generality for later reference, and will typically refer to some convex body.

Definition 2.1.8. For any subset A of some set $S \subset \mathbb{R}^d$, the **normal cone** of A is

$$NC_S(A) := \{y \in \mathbb{R}^d : x \cdot y \geq z \cdot y \quad \forall x \in A, z \in S\}.$$

Let $P \subset \mathbb{R}^d$ be some polytope. Then any face F will have a normal cone; in fact, it is well-known that the dimension of the normal cone for any face will directly relate to the dimension of the face, and that its structure will be fully determined by the geometry of the face.

Lemma 2.1.9. *Let $F \subset P$ be a k-dimensional face of a polytope $P \subset \mathbb{R}^d$.*

(a) $NC_P(F)$ has dimension $d - k$.

(b) Let $\mathbf{y}_1, ..., \mathbf{y}_m$ be the outward-directed normal vectors of every facet of P that contains F. Then

$$NC_P(F) = \{\lambda_1 \mathbf{y}_1 + ... + \lambda_m \mathbf{y}_m : \lambda_i \geq 0 \quad \forall 1 \leq i \leq m\}.$$

The normal cone of any face F consists of the linear combinations with non-negative coefficients of the normal vectors to the facets containing F. Note that in the case where P is a d-dimensional the normal cone of any facet is generated solely by its own outward-directed normal vector. By "outward-directed," we mean that if a normal vector were to begin on the facet, it would point out of P rather than into the interior of P. We can also alternatively discuss normal cones of faces in a polytope from the perspective of the vertices of the polytope.

Lemma 2.1.10. *Let v be a vertex of some $P \subset R^d$. Then:*

(a) *$NC_P(v)$ is a d-dimensional cone, consisting of the collection of all linear combinations with non-negative coefficients of the normal vectors of the facets containing v.*

(b) *For any face $F \subset P$, the normal cone of F is equal to the intersection of the normal cones of every vertex incident on F.*

Note that this means if the polytope P is d-dimensional, the normal cone of P itself will be the origin. Based on this concept of the normal cones, we can construct an object called the *normal fan*, which for any polytope will subdivide $\mathbb{R}^d$ according to the normal cones of its faces.

Definition 2.1.11. The **normal fan** of a polytope $P \subset \mathbb{R}^d$ is the collection of normal cones of the faces of the polytope:

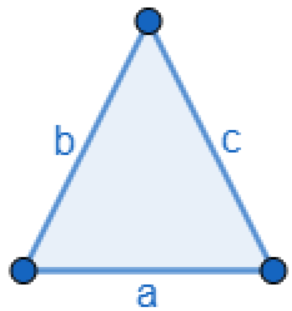
$$NF := \{NC_P(F) \subset \mathbb{R}^d : F \text{ is a face of } P\}$$

The structure of normal cones and normal fans, and how these relate to a polytope, can be directly visualized in $\mathbb{R}^2$.

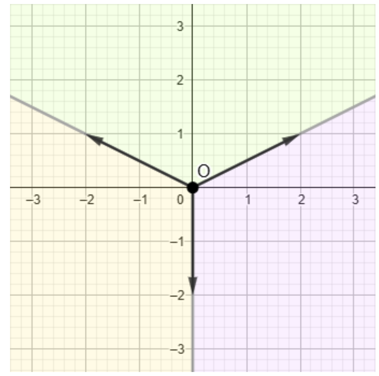
Example 2.1.12. Let P be the triangle shown in Figure 2.2a with edges labelled a, b, c. In $\mathbb{R}^2$, the facets of a polytope are simply the edges, and each edge lies on a bounding inequality for the polytope. However, up to rotation of P, the normal directions of each edge are independent of the exact location of P in $\mathbb{R}^2$:

$$\mathbf{v}_a = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \mathbf{v}_b = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \quad \mathbf{v}_c = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Each vertex of the triangle lies on exactly two facets (edges), and its two-dimensional normal cone is the set of all linear combinations with non-negative coefficients of



(a) A triangle in $\mathbb{R}^2$



(b) The normal fan for Figure 2.2a

Figure 2.2: The normal fan of a triangle in $\mathbb{R}^2$ includes three cones (green, orange, purple) which are the normal cones of the vertices, three rays in the normal direction of each edge which are the normal cones of the edges, and the origin.

the two normal vectors to its associated facets. The full normal fan of the triangle, shown in Figure 2.2b, consists of these three cones for the three vertices, the three rays outward from the origin that are the normal cones of the three edges, and the origin itself which is the normal cone of the full triangle.

One additional construction that we will use is the 1-*skeleton* of a polytope, which can be used to efficiently represent the edge structure.

Definition 2.1.13. The **1-skeleton** of any polytope $P \subset \mathbb{R}^d$ is the graph $G = (V, E)$ whose vertex set V is the vertices of the polytope and whose edge set E is the edge set of the polytope.

In essence, considering the 1-skeleton allows us to ignore the higher-dimensional structure of a polytope and focus on the vertices and edges alone. This is valuable in circumstances where the vertices have a separate combinatorial interpretation, as is the case with the family of polytopes introduced in Section 2.3. In such situations, the question of whether patterns can be observed in the polytope's edge structure, what those patterns might be, and how they relate to the combinatorial interpretation of the vertices is crucial to building up an understanding of the overall geometry. Ideally, the structure of the polytopes should depend on the combinatorial relationships among the objects represented by the vertices.

2.2 Partitions and the Partition Lattice

An *up-to-k-partition* $\mathcal{P}$ of n objects is a collection of k non-intersecting sets $B_1, ..., B_k \subset [n]$ called *blocks*, some of which may be empty, whose union is $[n]$. For simplicity, we restrict $k \leq n$. The Stirling numbers of the second kind $\left\{ {n \atop k} \right\}$ give the number of ways to partition n objects into k unordered non-empty blocks, meaning that the total number of up-to-k-partitions on a set of n objects is given by:

$$\sum_{i=1}^{k} \left\{ {n \atop i} \right\}$$

Here, partitions are considered to be unordered. To distinguish between classes of partitions on $[n]$, we will refer to partitions with exactly m nonempty blocks as m-block partitions. Specific categories of partitions will be identified by the sizes of their nonempty blocks, $b_1 \geq b_2 \geq ... \geq b_m$, and referred to as "$b_1 - b_2 - ... - b_m$ partitions;" for example, the partition $\{124\}, \{3\}$ of $[4]$ is a 2-block partition that falls into the class of 3-1 partitions.

One way to discuss relationships between partitions is through the concepts of *coarsening* and *refinement*.

Definition 2.2.1. A partition $\mathcal{P}$ is a **coarsening** of another partition $\mathcal{P}'$, in which case $\mathcal{P}'$ is a **refinement** of $\mathcal{P}$, if every block of $\mathcal{P}'$ is a subset of some block of $\mathcal{P}$.

Not all partitions of $[n]$ will be related to each other in this way; no m-block partition can be a coarsening or refinement of any other m-block partition, but even partitions with different numbers of blocks need not be related. For example, the 2-block partition $\{12\}, \{34\}$ of $[4]$ is not a coarsening of the 3-block partition $\{13\}, \{2\}, \{4\}$. However, this relationship does impose a partial order on the set of partitions of $[n]$. For $k = n$, the resulting poset is a lattice:

Definition 2.2.2. The **partition lattice** for a set $[n]$ is the lattice formed by the collection of all partitions on $[n]$, ordered by refinement: $\mathcal{P} \geq \mathcal{P}'$ if $\mathcal{P}'$ refines $\mathcal{P}$.

The top element of this lattice is the 1-block partition $\{123...n\}$, and the bottom element is the n-block partition $\{1\}, \{2\}, ..., \{n\}$. A diagram of this lattice for $n = 4$ is given in Figure 2.3. Edges in the diagram represent *cover* relationships.

Definition 2.2.3. An element $\mathcal{P}$ in a lattice is said to **cover** another element $\mathcal{P}'$ if $\mathcal{P} \geq \mathcal{P}'$ and there is no other partition $\mathcal{P}''$ such that $\mathcal{P} \geq \mathcal{P}'' \geq \mathcal{P}'$, i.e. there are no elements between $\mathcal{P}$ and $\mathcal{P}'$ in the ordering.

For example, in the lattice shown in Figure 2.3, $\{123\}, \{4\}$ covers $\{12\}, \{3\}, \{4\}$, and $\{1234\}$ covers $\{123\}, \{4\}$. However, although $\{1234\}$ is a coarsening of $\{12\}, \{3\}, \{4\}$, it does not cover $\{12\}, \{3\}, \{4\}$, because $\{123\}, \{4\}$ sits between them as a coarsening of $\{12\}, \{3\}, \{4\}$ and a refinement of $\{1234\}$.

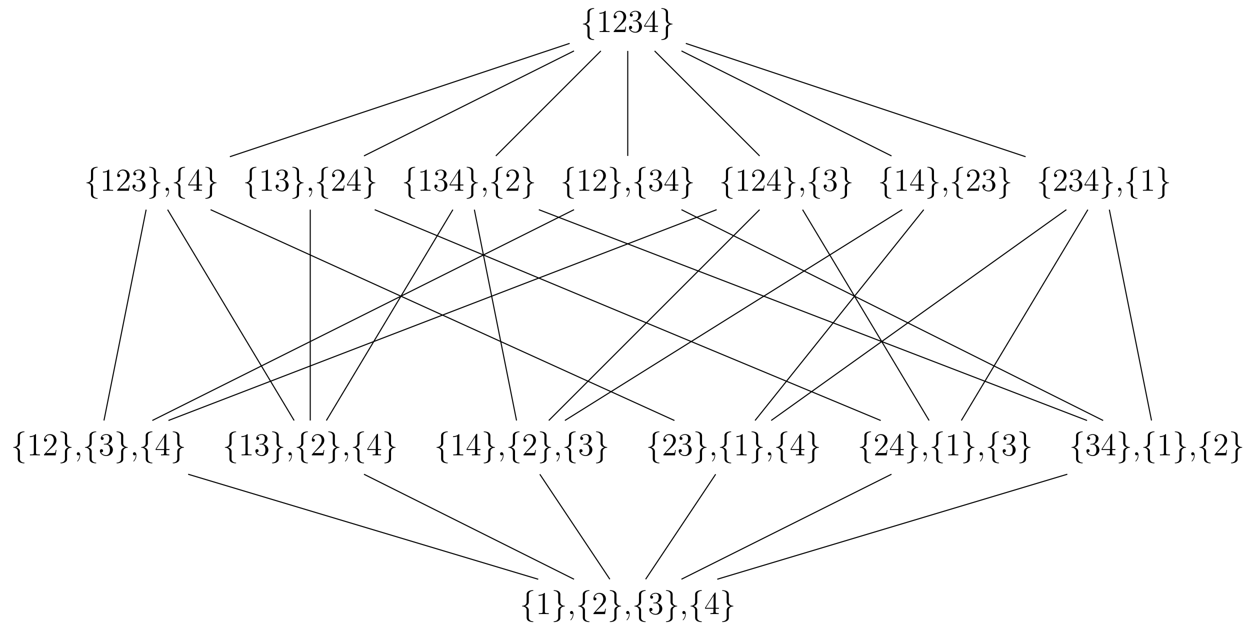


Figure 2.3: The Hasse diagram of the partition lattice for $n = 4$, ordered by refinement.

2.3 Up-to-k-Partition Polytopes

For a fixed n, k, we can represent each up-to-k-partition as an n-by-n symmetric matrix G, where $G_{ij} = 1$ if objects i and j are in the same block in the corresponding partition and $G_{ij} = -\frac{1}{k-1}$ otherwise. Note that $G_{ii} = 1$ for all $i = 1, ..., n$. From these symmetric matrices representing our partitions, we can obtain a unique point in $\mathbb{R}^{\binom{n}{2}}$ for each partition by pulling out the $\binom{n}{2}$ distinct elements. We construct this vector by taking each entry G_{ij} with $j > i$, moving across rows left-to-right and progressing downward from the first row. For $n = 4$, the procedure for constructing such a vector in $\binom{4}{2} = 6$ dimensions from a given matrix is shown below:

$$\begin{pmatrix} 1 & G_{12} & G_{13} & G_{14} \\ G_{12} & 1 & G_{23} & G_{24} \\ G_{13} & G_{23} & 1 & G_{34} \\ G_{14} & G_{24} & G_{34} & 1 \end{pmatrix} \longrightarrow (G_{12}, G_{13}, G_{14}, G_{23}, G_{24}, G_{34}) \in \mathbb{R}^6$$

By considering the collection of all such vectors generated by the up-to-k-partitions for a fixed n, k, we can construct a polytope.

Definition 2.3.1. For a fixed n and $k \leq n$, let V be the set of vectors in $\mathbb{R}^{\binom{n}{2}}$ generated by up-to-k-partitions of $[n]$. Then the **up-to-k-partition polytope** $L_{n,k}$ is the polytope obtained by taking the convex hull of the set V.

Note that these polytopes are defined using an integer n but exist in dimension $d = \binom{n}{2}$. This means the dimension in which the polytopes reside will scale on the order of n^2. In each space of the form $\mathbb{R}^d, d = \binom{n}{2}$, e.g. $d = 3, 6, 10, 15...$, a collection of $n - 1$ polytopes will exist: one for each value of $2 \leq k \leq n$. In any space $\mathbb{R}^d$ where $d \neq \binom{n}{2}$, no up-to-k-partition polytopes will exist.

It is interesting and useful to observe that the partition matrices are all elements in the space of *Gram matrices* of n unit vectors in $\mathbb{R}^n$. We will exclusively use the Euclidean inner product to define and discuss these matrices, but in general they can be endowed with any inner product.

Definition 2.3.2. The **Gram matrix** for any set of n vectors $\mathbf{v_1}, ..., \mathbf{v_n} \in \mathbb{R}^n$ is the n-by-n matrix G whose entries G_{ij} are given by

$$G_{ij} = \mathbf{v_i} \cdot \mathbf{v_j}$$

where $\mathbf{v_i} \cdot \mathbf{v_j}$ is the standard Euclidean inner product.

Note that any Gram matrix will be symmetric and positive semidefinite. In our setting, we focus on matrices generated by a set of n unit vectors, for which the entries along the main diagonal of any Gram matrix will always be $G_{ii} = 1, 1 \leq i \leq n$.

The partition matrices have a direct interpretation in how the n unit vectors are arranged with respect to each other in $\mathbb{R}^n$. Each partition corresponds to a vector arrangement in which objects in the same block correspond to identical vectors, and objects in different blocks are vectors evenly spaced away from one another, such that the inner product of any two vectors in different blocks will be $-\frac{1}{k-1}$. Then the inner product of any two vectors in the same block will be 1.

Example 2.3.3. Consider $n = 3$ and a trio of unit vectors $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3} \in \mathbb{R}^3$. The 1-block partition $\{123\}$ indicates that all three unit vectors are in the same block, i.e. all three are identical to each other, as in the top left of Figure 2.4. For $k = 2$, there are two available blocks, and the 2-block partition $\{13\}, \{2\}$ will place $\mathbf{v_1}, \mathbf{v_3}$ in a single block and $\mathbf{v_2}$ in a separate block. The farthest apart these blocks can get from each other is to point in opposite directions, as shown in the top right of Figure 2.4. For $k = 3$, the partition $\{13\}, \{2\}$ does not change, but the number of available blocks is now 3, meaning one block is now left empty. Therefore the vectors are arrayed to leave an "empty" slot for the blank third block, spaced evenly from the other two, as shown in the bottom left of Figure 2.4. This means that the spacing can remain constant when switching from a 2-block partition to the 3-block partition $\{1\}, \{2\}, \{3\}$, for which each of the three vectors must occupy a different block as in the bottom right of Figure 2.4.

By the construction of these polytopes, every point contained in each polytope will be a Gram matrix for some arrangement of n unit vectors; however, none of these polytopes is the complete space of Gram matrices on n unit vectors. That space has a more unusual convex geometric structure, and will be discussed more in Chapter 5. Moreover, by construction, the Gram matrices representing the partitions themselves will be the vertices of each partition polytope [34].

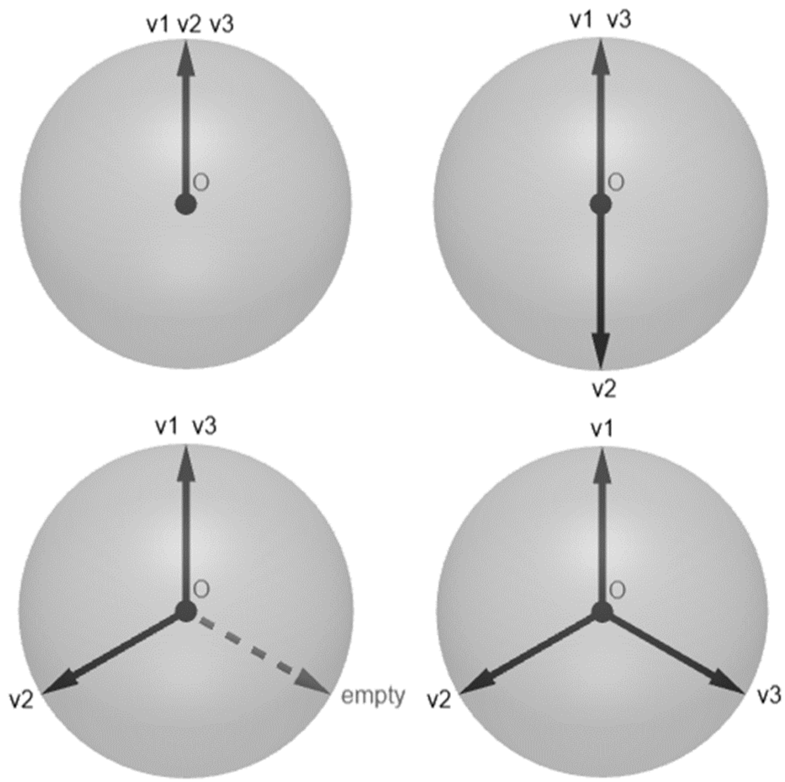


Figure 2.4: Top left: To correspond to the 1-block partition, all three vectors in the $n = 3$ set are identical and occupy a single block. Top right: For $k = 2$, a 2-block partition will correspond to one vector antiparallel to the other two. Bottom left: For $k = 3$, a 2-block partition means there are three equally-spaced locations, but one is left empty. Bottom right: Thanks to the empty block, the 3-block partition in $k = 3$ has a similar arrangement structure to the 2-block partition.

Example 2.3.4. One feasible partition of $[n]$ for $n = 4, k = 3$ is the 2-2 partition $\{12\}, \{34\}$ since it has fewer than k blocks. Using $k = 3$, this partition becomes the following matrix and point, which is then a vertex of $L_{4,3}$:

$$\{12\}, \{34\} \to \begin{pmatrix} 1 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ 1 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 & 1 \\ -\frac{1}{2} & -\frac{1}{2} & 1 & 1 \end{pmatrix} \longrightarrow \left(1, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, 1\right) \in \mathbb{R}^6$$

2.3.1 Relevant Past Work

Partition polytopes have been investigated from a variety of perspectives in the literature. Traditional work around them has involved a framework inspired by the max-cut problem on a complete graph, leading to their alternative naming as *cut polytopes*. Specifically, the polytope $L_{n,k}$ is the k-cut polytope for the complete graph K_n under an affine transformation, meaning its face lattice and combinatorial geometry are identical to the k-cut polytope. Past papers examining cut polytopes typically use 0/1 vertex coordinates, where a 0 indicates elements in the same block and a 1 indicates elements in different blocks, as opposed to the $\{1, -\frac{1}{k-1}\}$ coordinate structure described in Definition 2.3.1. Our alternative scaling to define $L_{n,k}$ is employed to ensure that the vertex set is identical to the vertex set of a convex spectrahedron known as the k-way elliptope, the significance of which will be discussed in Chapter 5.

Because the dimension of up-to-k-partition polytopes grows so quickly in n, they are difficult to model and evaluate in full. Their vertex descriptions are inherent to the construction, but their hyperplane descriptions are tricky to establish in generality, and much work has gone into elucidating these structures ([6], [34], [19], [22], [24], [46], [25]). Barahona and Mahjoub examine cut polytopes for generalized graphs, focused on how relationships between graphs foster relationships between their cut polytopes [6]. A result which will be helpful in this investigation is the following, which in the paper is a corollary of a more general statement about graph adjacencies:

Lemma 2.3.5. *[6] If $G = (V, E)$ is a complete graph, then the diameter of its 2-cut polytope is 1.*

This means that for any n, every vertex of $L_{n,2}$ is adjacent to every other vertex in $L_{n,2}$. Grötschel and Wakabayashi focus on facets of the k-cut polytopes for complete graphs, and establish that in the k-cut setting, every non-negativity constraint $x_i \geq 0$ is facet-defining across all k-cut polytopes [34]. They also characterize some facet-defining inequalities for 2-cut polytopes, as well as identify facets corresponding to regular substructures within the complete graph. Chopra similarly considers k-cut polytopes broadly, but focuses on characterizing facets to extend the work of Grötschel and Wakabayashi [19]. Deza and Laurent compile the most extensive results for the cut polytope across a number of papers ([22], [24], [46], among many others) as well as a book, *Geometry of Cuts and Metrics* [25]; these studies, too, focus on facet characterization for the polytopes. In one such paper, Deza, Grötschel, and Laurent fully describe the facet-defining inequalities for the $n = 5$ family of k-cut polytopes, identifying several categories of inequalities that could extend to higher dimensions [23]. Much of the current work in facet description is focused on generalizing these inequalities.

Beyond the results described above, the edge structure and other low-dimensional geometry of up-to-k-partition polytopes has been largely unstudied. The results presented in Chapter 3 and Chapter 4 will focus primarily on the edge structure of these polytopes, and will additionally make some progress towards an inductive approach to understanding and modeling the polytopes based on low-dimensional geometry. This is a particularly vital line of investigation for up-to-k-partition polytopes due to how their dimension changes with n. In other families of polytopes, such as hypercubes, facets of a d-dimensional polytope can be $(d-1)$-dimensional members of that same family; however, the up-to-k-partition polytopes skip several dimensions with each incremental change in n, meaning that their facets cannot also be partition polytopes. Our main results, presented in Chapter 4, establish that the up-to-k-partition polytopes for $n-1$ exist as $\binom{n-1}{2}$-dimensioned faces of $L_{n,k}$ (Theorem 4.2.4) and moreover that this behavior persists recursively for even smaller polytopes.

Chapter Three

Case studies: Partition polytopes for $n = 4$ and $n = 5$

To investigate the low-dimensional structure of up-to-k-partition polytopes, we begin by inspecting some concrete examples to build intuition. The only polytopes that can be visualized fully are for $n = 3$, which gives 3-dimensional polytopes for both $k = 2$ and $k = 3$:

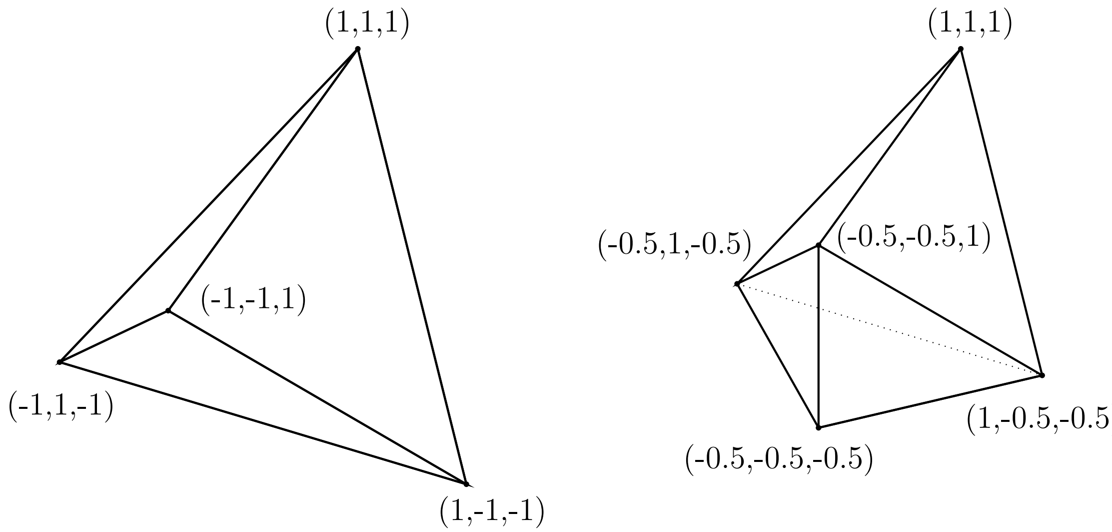


Figure 3.1: $L_{3,2}$ (left) and $L_{3,3}$ (right)

These examples are too small to glean much information about higher behavior, but they provide a good scaffold for observation and understanding. Recall from Definition 2.3.1 that the vertices of $L_{3,2}$ and $L_{3,3}$ represent the partitions of [3]. In $L_{3,2}$, only the 1-block and 2-block partitions are represented. One vertex represents the unique 1-block partition $\{123\}$, and the other three vertices represent the three distinct 2-block partitions. Every vertex in $L_{3,2}$ is adjacent to every other vertex. Keeping in mind that each vertex is derived from a special Gram matrix, where each entry G_{ij} represents the dot product of two unit vectors among a set of n unit vectors, the vertices and their associated matrices represent partitions as follows:

$$(1,1,1) \to \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \to \{123\}$$

$$(1,-1,-1) \to \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} \to \{12\},\{3\}$$

$$(-1,1,-1) \to \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix} \to \{13\},\{2\}$$

$$(-1,-1,1) \to \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \to \{23\},\{1\}$$

In $L_{3,3}$, there is one more vertex than $L_{3,2}$, and the new vertex represents the unique 3-block partition $\{1\},\{2\},\{3\}$. The three vertices representing 2-block partitions have shifted in accordance with the $\left\{1, -\frac{1}{k-1}\right\}$ paradigm given in the matrix construction prior to Definition 2.3.1:

$$\{12\},\{3\} \to \left(1, -\frac{1}{2}, -\frac{1}{2}\right)$$
$$\{13\},\{2\} \to \left(-\frac{1}{2}, 1, -\frac{1}{2}\right)$$
$$\{23\},\{1\} \to \left(-\frac{1}{2}, -\frac{1}{2}, 1\right)$$

Refer back to Figure 2.4 to see how these positions correspond to unit vector placement through the Gram matrix interpretation. The adjacencies possessed by the 2-block vertices in $L_{3,2}$ have not changed in $L_{3,3}$ – each is still adjacent to the 1-block vertex and to both of the other 2-block vertices. However, some new adjacencies have been introduced by the 3-block partition vertex. The new vertex is adjacent to each of the 2-block partition vertices, but it is not adjacent to the 1-block vertex. The 1-block vertex remains solely adjacent to the 2-block vertices.

From the $n = 3$ cases, we can immediately identify that two vertices are unusual: the 1-block partition vertex, which will remain constant for all k-values, and the n-block partition vertex, which appears only at $k = n$. We define both of these points for easy reference.

Definition 3.0.1. Let $L_{n,k}$ be an up-to-k-partition polytope. Then the **peak vertex** v_α is the vertex representing the unique 1-block partition, located at $(1, 1, ..., 1) \in \mathbb{R}^{\binom{n}{2}}$.

Definition 3.0.2. Let $L_{n,n}$ be an up-to-n-partition polytope. Then the **basal vertex** v_ω is the vertex representing the unique n-block partition, located at $\left(-\frac{1}{n-1}, -\frac{1}{n-1}, ..., -\frac{1}{n-1}\right) \in \mathbb{R}^{\binom{n}{2}}$.

Beyond this, there is not much to be observed from the $n = 3$ case, and so the bulk of this chapter will focus on higher-dimensional polytopes. Since the dimension of these polytopes expands on the order of n^2 as discussed in Section 2.3, only a few small examples can be fully rendered in Sage. What follows is a complete discussion of the polytopes that occur using $n = 4$ and $n = 5$ with each possible value of k. In general, vertices will be identified by the partitions they represent.

3.1 Methods

All models were constructed using Sage in the virtual computation service CoCalc [64]. Since the vertices of $L_{n,k}$ are known for any n, k, we can construct polytopes $L_{n,k}$ using the Polyhedron command in Sage, hardcoding the list of vertices whose convex hull will be taken to yield the polytope. There are technical limitations to this; over CoCalc, Sage could not construct any $n = 6$ polytope in a reasonable amount of time, limiting the case studies to $n = 4$ and $n = 5$. Within the case studies, Sage was used to compute the f-vectors of each polytope and establish whether they were simple, simplicial, and/or ℓ-neighborly for any $3 \leq \ell \leq \frac{n}{2}$. In addition, the facet-defining hyperplanes were computed for all $n = 4$ polytopes, and the full adjacency matrix for every polytope in both $n = 4$ and $n = 5$ was explored to identify patterns regarding which vertices are adjacent to each other within each $L_{n,k}$. Lastly, based on an observation that all but a few 6-dimensional faces of $L_{5,3}$ were simplices, the non-simplicial 6-dimensional faces of each $n = 5$ polytope were investigated for structural similarities to the 6-dimensional $(n = 4)$ $L_{n,k}$ polytopes.

3.2 Observations on $n = 4$

For the case $n = 4$, we vary k from 2 to 4 to get a trio of polytopes in $\binom{4}{2} = 6$ dimensions. Each vertex of a particular polytope corresponds to a unique partition with at most k non-empty blocks, with the location of that vertex having the form

$$(G_{1,2}, G_{1,3}, G_{1,4}, G_{2,3}, G_{2,4}, G_{3,4}) \in \mathbb{R}^6$$

corresponding to the entries of a unique Gram matrix. Each element is equal to either 1 or $-\frac{1}{k-1}$ to indicate that objects i and j are in the same block or different blocks, respectively, of the associated partition. Note that up to scaling, this is the standard up-to-k-partition polytope (or k-cut polytope) as examined in [6, 19, 24], with the only notable difference being which constant values are used to represent which relationships.

3.2.1 $L_{4,2}$

The f-vector for $L_{4,2}$ is $(1, 8, 28, 56, 68, 48, 16, 1)$, indicating that the polytope has 8 vertices, 28 edges, and 16 facets, as well 172 middle-dimensional faces distributed across several levels. We can observe that $28 = \binom{8}{2}$, meaning that all possible edges between pairs of the 8 vertices are present in this polytope. The sixteen facet-defining inequalities for this polytope, where $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)^T$ represents a column vector in $\mathbb{R}^6$, are:

$$(1, 1, 0, 1, 0, 0)\mathbf{x} \geq -1 \tag{3.1}$$

$$(1, 0, 1, 0, 1, 0)\mathbf{x} \geq -1 \tag{3.2}$$

$$(0, 0, 0, 1, 1, 1)\mathbf{x} \geq -1 \tag{3.3}$$

$$(0, 1, 1, 0, 0, 1)\mathbf{x} \geq -1 \tag{3.4}$$

$$(-1, -1, 0, 1, 0, 0)\mathbf{x} \geq -1 \tag{3.5}$$

$$(-1, 1, 0, -1, 0, 0)\mathbf{x} \geq -1 \tag{3.6}$$

$$(1, -1, 0, -1, 0, 0)\mathbf{x} \geq -1 \tag{3.7}$$

$$(-1, 0, -1, 0, 1, 0)\mathbf{x} \geq -1 \tag{3.8}$$

$$(-1, 0, 1, 0, -1, 0)\mathbf{x} \geq -1 \tag{3.9}$$

$$(1, 0, -1, 0, -1, 0)\mathbf{x} \geq -1 \tag{3.10}$$

$$(0, 0, 0, -1, -1, 1)\mathbf{x} \geq -1 \tag{3.11}$$

$$(0, 0, 0, -1, 1, -1)\mathbf{x} \geq -1 \tag{3.12}$$

$$(0, 0, 0, 1, -1, -1)\mathbf{x} \geq -1 \tag{3.13}$$

$$(0, -1, -1, 0, 0, 1)\mathbf{x} \geq -1 \tag{3.14}$$

$$(0, -1, 1, 0, 0, -1)\mathbf{x} \geq -1 \tag{3.15}$$

$$(0, 1, -1, 0, 0, -1)\mathbf{x} \geq -1 \tag{3.16}$$

$L_{4,2}$ is simplicial (Definition 2.1.5), meaning that each facet is a simplex and thus contains exactly 6 vertices, since the facets are 5-dimensional faces. $L_{4,2}$ is also 3-neighborly (Definition 2.1.6); every possible subset of 3 or fewer vertices forms a face of the polytope, i.e. every trio of vertices forms a 2-dimensional face, which aligns with the observation that the polytope has $56 = \binom{8}{3}$ 2-dimensional faces.

Of the 8 vertices for $L_{4,2}$, 1 is the peak vertex v_α, 4 represent 3-1 partitions, and 3 represent 2-2 partitions. Each 3-1 partition vertex has 3 positive entries and 3 negative, while the 2-2 partition vertices each have 2 positive entries and 4 negative, indicating which pairs of objects are in the same block of the corresponding partition.

$$\begin{aligned}
\text{3-1 partition vertices: } &\{123\}, \{4\} \to (1, 1, -1, 1, -1, -1);\\
&\{124\}, \{3\} \to (1, -1, 1, -1, 1, -1);\\
&\{134\}, \{2\} \to (-1, 1, 1, -1, -1, 1);\\
&\{234\}, \{1\} \to (-1, -1, -1, 1, 1, 1)\\
\text{2-2 partition vertices: } &\{12\}, \{34\} \to (1, -1, -1, -1, -1, 1);
\end{aligned}$$

$$\{13\},\{24\} \to (-1,1,-1,-1,1,-1);$$
$$\{14\},\{23\} \to (-1,-1,1,1,-1,-1)$$

3.2.2 $L_{4,3}$

The f-vector for $L_{4,3}$ is $(1, 14, 85, 222, 251, 123, 23, 1)$, indicating that the polytope has 14 vertices, 85 edges, and 23 facets, as well as 596 middle-dimensional faces, over three times as many as $L_{4,2}$. $L_{4,3}$ has six more vertices than $L_{4,2}$, with the new vertices representing the six partitions of 4 objects into 3 non-empty blocks. All of these are 2-1-1 partitions, with 5 negative coordinates and a single positive coordinate indicating the unique pair of objects joined together in the size-2 block of the corresponding partition. Moreover, the vertices representing the 2-block partitions have all shifted, with every -1 entry from their $L_{4,2}$ locations becoming $-\frac{1}{2}$ in the $L_{4,3}$ locations.

$$\begin{aligned}
\text{3-1 partition vertices: } \{123\},\{4\} &\to \left(1,1,-\frac{1}{2},1,-\frac{1}{2},-\frac{1}{2}\right);\\
\{124\},\{3\} &\to \left(1,-\frac{1}{2},1,-\frac{1}{2},1,-\frac{1}{2}\right);\\
\{134\},\{2\} &\to \left(-\frac{1}{2},1,1,-\frac{1}{2},-\frac{1}{2},1\right);\\
\{234\},\{1\} &\to \left(-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},1,1,1\right)\\
\text{2-2 partition vertices: } \{12\},\{34\} &\to \left(1,-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},1\right);\\
\{13\},\{24\} &\to \left(-\frac{1}{2},1,-\frac{1}{2},-\frac{1}{2},1,-\frac{1}{2}\right);\\
\{14\},\{23\} &\to \left(-\frac{1}{2},-\frac{1}{2},1,1,-\frac{1}{2},-\frac{1}{2}\right)\\
\text{2-1-1 partition vertices: } \{12\},\{3\},\{4\} &\to \left(1,-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2}\right);
\end{aligned}$$

$$\{13\},\{2\},\{4\} \to \left(-\frac{1}{2},1,-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2}\right);$$
$$\{14\},\{2\},\{3\} \to \left(-\frac{1}{2},-\frac{1}{2},1,-\frac{1}{2},-\frac{1}{2},-\frac{1}{2}\right);$$
$$\{23\},\{1\},\{4\} \to \left(-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},1,-\frac{1}{2},-\frac{1}{2}\right);$$
$$\{24\},\{1\},\{3\} \to \left(-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},1,-\frac{1}{2}\right);$$
$$\{34\},\{1\},\{2\} \to \left(-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2},1\right)$$

Not coincidentally, the polytope has 85 edges, which is precisely 6 shy of $\binom{14}{2} = 91$. On inspection, the absent edges are those which would connect the 6 new vertices to $v_\alpha = (1, 1, 1, 1, 1, 1)$. Thus v_α remains adjacent solely to the 2-block vertices, to which it was also adjacent in $L_{4,2}$. Otherwise, all vertices are adjacent to all other vertices: every 2-block partition vertex is adjacent to every other 2-block and 3-block partition vertex, and every 3-block partition vertex is adjacent to every other 2-block and 3-block partition vertex.

$L_{4,3}$ has 7 more facets than $L_{4,2}$, and it is not the case that every facet-defining inequality of $L_{4,2}$ is a facet-defining inequality of $L_{4,3}$. The 23 facet-defining inequalities for $L_{4,3}$ are:

$$(-1,-1,0,1,0,0)x \geq -1 \tag{3.17}$$
$$(-1,1,0,-1,0,0)x \geq -1 \tag{3.18}$$
$$(1,-1,0,-1,0,0)x \geq -1 \tag{3.19}$$
$$(-1,0,-1,0,1,0)x \geq -1 \tag{3.20}$$
$$(-1,0,1,0,-1,0)x \geq -1 \tag{3.21}$$
$$(1,0,-1,0,-1,0)x \geq -1 \tag{3.22}$$

$$(0,0,0,-1,-1,1)x \geq -1 \tag{3.23}$$

$$(0,0,0,-1,1,-1)x \geq -1 \tag{3.24}$$

$$(0,0,0,1,-1,-1)x \geq -1 \tag{3.25}$$

$$(0,-1,-1,0,0,1)x \geq -1 \tag{3.26}$$

$$(0,-1,1,0,0,-1)x \geq -1 \tag{3.27}$$

$$(0,1,-1,0,0,-1)x \geq -1 \tag{3.28}$$

$$(1,0,0,0,0,0)x \geq -\frac{1}{2} \tag{3.29}$$

$$(0,1,0,0,0,0)x \geq -\frac{1}{2} \tag{3.30}$$

$$(0,0,1,0,0,0)x \geq -\frac{1}{2} \tag{3.31}$$

$$(0,0,0,1,0,0)x \geq -\frac{1}{2} \tag{3.32}$$

$$(0,0,0,0,1,0)x \geq -\frac{1}{2} \tag{3.33}$$

$$(0,0,0,0,0,1)x \geq -\frac{1}{2} \tag{3.34}$$

$$(1,1,-1,1,-1,-1)x \geq -\frac{3}{2} \tag{3.35}$$

$$(1,-1,1,-1,1,-1)x \geq -\frac{3}{2} \tag{3.36}$$

$$(-1,-1,-1,1,1,1)x \geq -\frac{3}{2} \tag{3.37}$$

$$(-1,1,1,-1,-1,1)x \geq -\frac{3}{2} \tag{3.38}$$

$$(1,1,1,1,1,1)x \geq -\frac{3}{2} \tag{3.39}$$

In particular, the first four inequalities for $L_{4,2}$ (Equations 3.1-3.4) are no longer present in the list of facet-defining inequalities for $L_{4,3}$. Constraints of $x_i \geq -\frac{1}{2}$ have been added for $i = 1, ..., 6$. This aligns with the broad construction for $L_{4,k}$ based on partition vertices as described in Section 2.3 and particularly Definition 2.3.1; by design, each partition vertex has entries in $\{1, -\frac{1}{k-1}\}$, meaning that for $k = 3$ no coordinate of any vertex can go below $-\frac{1}{k-1} = -\frac{1}{2}$. Five additional new constraints have also appeared. This polytope is no longer simplicial or 3-neighborly; its geometry is more intricate than $L_{4,2}$.

3.2.3 $L_{4,4}$

The f-vector for $L_{4,4}$ is $(1, 15, 88, 218, 240, 117, 22, 1)$, indicating that the polytope has 15 vertices, 88 edges, and 22 facets, as well as 575 middle-dimensional faces. One new vertex has been added to the partitions represented in $L_{4,3}$: the vertex representing the unique partition of 4 objects into exactly 4 blocks, i.e. the basal vertex v_ω. The vertex v_ω is adjacent exactly to the collection of 3-block partition vertices, giving it 6 edges in total. Meanwhile, 3 edges present in $L_{4,3}$ are absent – there are no longer edges connecting the vertices representing the partitions $\{1, 2, 34\}$ and $\{12, 3, 4\}$, the partitions $\{1, 3, 24\}$ and $\{13, 2, 4\}$, and the partitions $\{1, 4, 23\}$ and $\{14, 2, 3\}$. We can observe that the 3-block partitions which have been separated are pairs of partitions whose blocks of size 2 do not share an element. The partition vertices which were present in $L_{4,2}$ and $L_{4,3}$ are now located at coordinates with the same sign patterns as in $L_{4,2}$ and $L_{4,3}$, but with the negative values all being $-\frac{1}{3}$; the new basal vertex is located as shown below.

$$\begin{aligned}
\text{3-1 partition vertices: } \{123\}, \{4\} &\to \left(1, 1, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}\right); \\
\{124\}, \{3\} &\to \left(1, -\frac{1}{3}, 1, -\frac{1}{3}, 1, -\frac{1}{3}\right); \\
\{134\}, \{2\} &\to \left(-\frac{1}{3}, 1, 1, -\frac{1}{3}, -\frac{1}{3}, 1\right); \\
\{234\}, \{1\} &\to \left(-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, 1, 1\right) \\
\text{2-2 partition vertices: } \{12\}, \{34\} &\to \left(1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1\right);
\end{aligned}$$

$$\begin{aligned}
&\{13\},\{24\} \to \left(-\frac{1}{3},1,-\frac{1}{3},-\frac{1}{3},1,-\frac{1}{3}\right);\\
&\{14\},\{23\} \to \left(-\frac{1}{3},-\frac{1}{3},1,1,-\frac{1}{3},-\frac{1}{3}\right)\\
\text{2-1-1 partition vertices: }&\{12\},\{3\},\{4\} \to \left(1,-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3}\right);\\
&\{13\},\{2\},\{4\} \to \left(-\frac{1}{3},1,-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3}\right);\\
&\{14\},\{2\},\{3\} \to \left(-\frac{1}{3},-\frac{1}{3},1,-\frac{1}{3},-\frac{1}{3},-\frac{1}{3}\right);\\
&\{23\},\{1\},\{4\} \to \left(-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},1,-\frac{1}{3},-\frac{1}{3}\right);\\
&\{24\},\{1\},\{3\} \to \left(-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},1,-\frac{1}{3}\right);\\
&\{34\},\{1\},\{2\} \to \left(-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},1\right)\\
\text{1-1-1-1 partition vertex } v_\omega\text{: }&\{1\},\{2\},\{3\},\{4\} \to \left(-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},-\frac{1}{3}\right)
\end{aligned}$$

As for the facets, this polytope has one facet fewer than $L_{4,3}$, and its facet inequalities are otherwise highly similar to those of $L_{4,3}$. The first 12 inequalities of $L_{4,3}$ (Equations 3.17 – 3.28) are also facet-defining for $L_{4,4}$. The next 10 inequalities of $L_{4,3}$ can be shifted by a constant value to obtain facet-defining inequalities for $L_{4,4}$. The inequalities involving $-\frac{1}{2}$ for $L_{4,3}$, which are Equations 3.29 – 3.34, now use $-\frac{1}{3}$; those involving $-\frac{3}{2}$ for $L_{4,3}$, which are Equations 3.35 – 3.38, now use $-\frac{4}{3}$. These adjustments reflect the shift in structure from $L_{4,3}$ to $L_{4,4}$, since the vertices for each polytope are specified using k. Up to this change in constants, the inequalities are identical in structure. The new versions of these 10 altered inequalities are given below.

$$(1,0,0,0,0,0)x \geq -\frac{1}{3} \tag{3.40}$$

$$(0,1,0,0,0,0)x \geq -\frac{1}{3} \tag{3.41}$$

$$(0,0,1,0,0,0)x \geq -\frac{1}{3} \tag{3.42}$$

$$(0,0,0,1,0,0)x \geq -\frac{1}{3} \tag{3.43}$$

$$(0,0,0,0,1,0)x \geq -\frac{1}{3} \tag{3.44}$$

$$(0,0,0,0,0,1)x \geq -\frac{1}{3} \tag{3.45}$$

$$(1,1,-1,1,-1,-1)x \geq -\frac{4}{3} \tag{3.46}$$

$$(1,-1,1,-1,1,-1)x \geq -\frac{4}{3} \tag{3.47}$$

$$(-1,-1,-1,1,1,1)x \geq -\frac{4}{3} \tag{3.48}$$

$$(-1,1,1,-1,-1,1)x \geq -\frac{4}{3} \tag{3.49}$$

The final facet-defining inequality listed for $L_{4,3}$, Equation 3.39, is absent from $L_{4,4}$. The basal vertex sits precisely at the intersection of the 6 inequalities of the form $x_i \geq -\frac{1}{3}$, since it represents a partition where all elements are in separate blocks.

We can directly compare the 1-skeleton of $L_{4,4}$ as depicted in Figure 3.2 to the $n = 4$ partition lattice shown in Figure 2.3. On close inspection, it can be seen that every edge in the $n = 4$ partition lattice is also an edge of $L_{4,4}$. However, it is also clear that the 1-skeleton of $L_{4,4}$ is substantially more interconnected than the $n = 4$ partition lattice. The vertices representing the 2-block and 3-block partitions nearly form a complete graph, with only the few absent edges described previously. This suggests that the partition lattice relationships may be involved in determining some fraction of the edge structure of $L_{n,n}$, or more broadly $L_{n,k}$, but there must also be a wide variety of other factors by which edges can be identified in these polytopes.

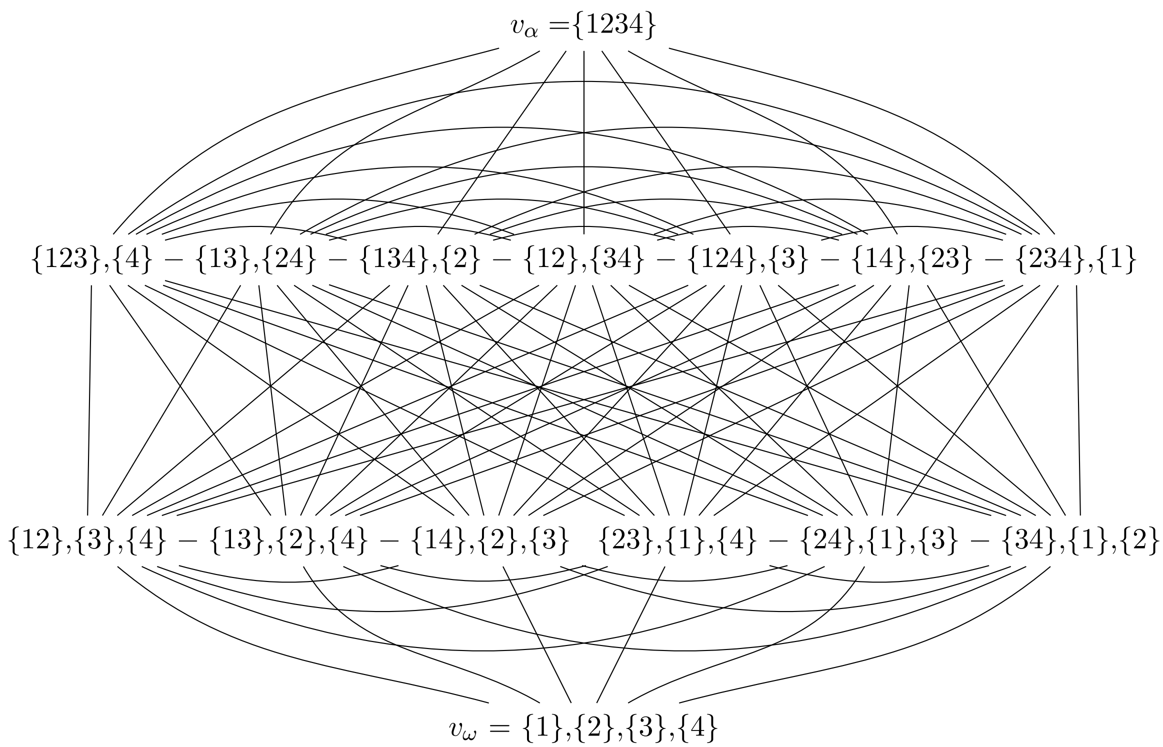


Figure 3.2: The 1-skeleton graph of $L_{4,4}$, with edges in the polytope represented by edges in the graph.

3.3 Observations on $n = 5$

For the case $n = 5$ and any k, the polytopes are located in $\mathbb{R}^{\binom{5}{2}} = \mathbb{R}^{10}$. Each vertex of a particular polytope will correspond to a unique partition with at most k non-empty blocks, and will take the form

$$(G_{1,2}, G_{1,3}, G_{1,4}, G_{1,5}, G_{2,3}, G_{2,4}, G_{2,5}, G_{3,4}, G_{3,5}, G_{4,5}) \in \mathbb{R}^{10}$$

where each element of the vertex coordinate string denotes whether object i and object j are in the same block in the corresponding partition. The set of vertices for a particular polytope, determined by k, represents the full set of partitions on n objects with up to k non-empty blocks.

These polytopes have many more facets than the $n = 4$ polytopes, so instead of providing a full listing of the facet inequalities, the f-vectors will be listed here for reference and detailed observations related to each polytope will be handled separately.

$$L_{5,2} : (1, 16, 120, 560, 1780, 3872, 5592, 5060, 2600, 640, 56, 1)$$

$$L_{5,3} : (1, 41, 795, 8070, 35790, 77774, 89482, 56825, 20205, 3895, 333, 1)$$

$$L_{5,4} : (1, 51, 1070, 9060, 34385, 67416, 73244, 45140, 15655, 2930, 243, 1)$$

$$L_{5,5} : (1, 52, 1065, 8990, 34200, 67169, 73034, 45020, 15610, 2920, 242, 1)$$

From the f-vectors, we can see that there are also too many vertices to effectively list them individually for each polytope. Instead, we will make use of a property that can be observed in the $n = 4$ case: by construction, the *sign pattern* for any m-block partition vertex will be consistent across all polytopes $L_{n,k}$ for $m \leq k \leq n$.

Example 3.3.1. In $L_{4,2}$, the partition vertex representing $\{123\}, \{4\}$ is:

$$(1, 1, -1, 1, -1, -1)$$

The corresponding sign pattern replaces every positive value with + and every negative value with − to get:

$$(+, +, -, +, -, -)$$

To get the vertices representing $\{123\}, \{4\}$ in $L_{4,3}$ and $L_{4,4}$, the -1 values in the partition representation are replaced with $-\frac{1}{2}$ and $-\frac{1}{3}$, respectively, which are still negative. Thus these vertices have the same sign pattern:

$$\left(1, 1, -\frac{1}{2}, 1, -\frac{1}{2}, -\frac{1}{2}\right) \to (+, +, -, +, -, -)$$
$$\left(1, 1, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}\right) \to (+, +, -, +, -, -)$$

For each new value of k, we will give the sign patterns of the added vertices and state the entries which correspond to the positive and negative signs in the vertices of $L_{5,k}$.

3.3.1 $L_{5,2}$

This polytope has 16 vertices, one of which is the peak vertex v_α. The other 15 vertices represent the 2-block partitions, of which there are two categories: 4-1 partitions and 3-2 partitions. The five 4-1 partitions are vertices with 6 positive coordinates and 4 negatives, while the ten 3-2 partitions each have 4 positive and 6 negative. The sign patterns are given below. In $L_{5,2}$, a positive sign indicates a value of 1, while a negative sign indicates a value of -1.

$$
\begin{aligned}
v_\alpha\text{: } & \{12345\} \to (+,+,+,+,+,+,+,+,+,+) \\
\text{4-1 partitions: } & \{1234\},\{5\} \to (+,+,+,-,+,+,-,+,-,-) \\
& \{1235\},\{4\} \to (+,+,-,+,+,-,+,-,+,-) \\
& \{1245\},\{3\} \to (+,-,+,+,-,+,+,-,-,+) \\
& \{1345\},\{2\} \to (-,+,+,+,-,-,-,+,+,+) \\
& \{2345\},\{1\} \to (-,-,-,-,+,+,+,+,+,+) \\
\text{3-2 partitions: } & \{123\},\{45\} \to (+,+,-,-,+,-,-,-,-,+) \\
& \{124\},\{35\} \to (+,-,+,-,-,+,-,-,+,-) \\
& \{125\},\{34\} \to (+,-,-,+,-,-,+,+,-,-) \\
& \{134\},\{25\} \to (-,+,+,-,-,-,+,+,-,-) \\
& \{135\},\{24\} \to (-,+,-,+,-,+,-,-,+,-) \\
& \{145\},\{23\} \to (-,-,+,+,+,-,-,-,-,+) \\
& \{234\},\{15\} \to (-,-,-,+,+,+,-,+,-,-) \\
& \{235\},\{14\} \to (-,-,+,-,+,-,+,-,+,-) \\
& \{245\},\{13\} \to (-,+,-,-,-,+,+,-,-,+) \\
& \{345\},\{12\} \to (+,-,-,-,-,-,-,+,+,+)
\end{aligned}
$$

In $L_{5,2}$, all vertices are adjacent to each other – that is, all $\binom{16}{2} = 120$ possible edges are present in the polytope. Unlike $L_{4,2}$, $L_{5,2}$ is not simplicial. However, it is 3-neighborly, with every possible subset of 3 vertices forming a two-dimensional face; this can be verified using the f-vector by observing that the number of two-dimensional faces in $L_{5,2}$ is $560 = \binom{16}{3}$.

3.3.2 $L_{5,3}$

$L_{5,3}$ has very regular but intricate adjacency behavior. There are 41 vertices in this polytope. The 25 additional vertices compared to $L_{5,2}$ represent 3-1-1 partitions and 2-2-1 partitions, and their sign patterns are given below. Each 3-1-1 partition vertex has 3 positive and 7 negative coordinates, while each 2-2-1 partition vertex has 2 positive and 8 negative coordinates. The sign patterns of v_α and the 15 vertices representing 2-block partitions, which were also represented in $L_{5,2}$, remain constant. In $L_{5,3}$, a positive sign still indicates a value of 1, but a negative sign now indicates a value of $-\frac{1}{2}$.

$$
\begin{aligned}
\text{3-1-1 partitions: } & \{123\},\{4\},\{5\} \to (+,+,-,-,+,-,-,-,-,-)\\
& \{124\},\{3\},\{5\} \to (+,-,+,-,-,+,-,-,-,-)\\
& \{125\},\{3\},\{4\} \to (+,-,-,+,-,-,+,-,-,-)\\
& \{134\},\{2\},\{5\} \to (-,+,+,-,-,-,-,+,-,-)\\
& \{135\},\{2\},\{4\} \to (-,+,-,+,-,-,-,-,+,-)\\
& \{145\},\{2\},\{3\} \to (-,-,+,+,-,-,-,-,-,+)\\
& \{234\},\{1\},\{5\} \to (-,-,-,-,+,+,-,+,-,-)\\
& \{235\},\{1\},\{4\} \to (-,-,-,-,+,-,+,-,+,-)\\
& \{245\},\{1\},\{3\} \to (-,-,-,-,-,+,+,-,-,+)\\
& \{345\},\{1\},\{2\} \to (-,-,-,-,-,-,-,+,+,+)\\
\text{2-2-1 partitions: } & \{12\},\{34\},\{5\} \to (+,-,-,-,-,-,-,+,-,-)\\
& \{13\},\{24\},\{5\} \to (-,+,-,-,-,+,-,-,-,-)\\
& \{14\},\{23\},\{5\} \to (-,-,+,-,+,-,-,-,-,-)\\
& \{12\},\{35\},\{4\} \to (+,-,-,-,-,-,-,-,+,-)
\end{aligned}
$$

$$\{13\},\{25\},\{4\} \to (-,+,-,-,-,-,+,-,-,-)$$

$$\{15\},\{23\},\{4\} \to (-,-,-,+,+,-,-,-,-,-)$$

$$\{12\},\{45\},\{3\} \to (+,-,-,-,-,-,-,-,-,+)$$

$$\{14\},\{25\},\{3\} \to (-,-,+,-,-,-,+,-,-,-)$$

$$\{15\},\{24\},\{3\} \to (-,-,-,+,-,+,-,-,-,-)$$

$$\{13\},\{45\},\{2\} \to (-,+,-,-,-,-,-,-,-,+)$$

$$\{14\},\{35\},\{2\} \to (-,-,+,-,-,-,-,-,+,-)$$

$$\{15\},\{34\},\{2\} \to (-,-,-,+,-,-,-,+,-,-)$$

$$\{23\},\{45\},\{1\} \to (-,-,-,-,+,-,-,-,-,+)$$

$$\{24\},\{35\},\{1\} \to (-,-,-,-,-,+,-,-,+,-)$$

$$\{25\},\{34\},\{1\} \to (-,-,-,-,-,-,+,+,-,-)$$

Every vertex corresponding to a 2-block partition is adjacent to all other vertices. Every vertex corresponding to a 3-block partition is adjacent to all other vertices *except* the peak vertex v_α. Meanwhile, v_α is adjacent to all vertices representing 2-block partitions. This is identical to the behavior observed in $L_{4,3}$ in Subsection 3.2.2, in which v_α is adjacent solely to the 2-block vertices, and the set of vertices excluding v_α are all adjacent to one another. As in that case, we can further confirm this with the observation that the number of edges listed for $L_{5,3}$ in the f-vector is $795 = \binom{41}{2} - 25$, corresponding to the absence of the 25 edges that would otherwise connect each 3-block partition vertex to v_α. In the next chapter, we will prove that in fact v_α is exactly adjacent to the set of 2-block partition vertices for any values of n and k.

3.3.3 $L_{5,4}$

There are 51 vertices in $L_{5,4}$, and at this stage their interactions are much more complex. The 10 new vertices represent the 4-block partitions, all of which are 2-1-1-1 partitions, and each has a single positive coordinate accompanied by 9 negative coordinates. The single positive coordinate represents the size-2 block in each partition. As before, the sign patterns of the 41 vertices representing partitions with fewer blocks are the same as they were in $L_{5,3}$. In $L_{5,4}$, a positive entry in the sign pattern indicates a value of 1, and a negative entry indicates a value of $-\frac{1}{3}$.

$$
\begin{array}{rl}
\text{2-1-1-1 partitions:} & \{12\},\{3\},\{4\},\{5\} \rightarrow (+,-,-,-,-,-,-,-,-,-) \\
& \{13\},\{2\},\{4\},\{5\} \rightarrow (-,+,-,-,-,-,-,-,-,-) \\
& \{14\},\{2\},\{3\},\{5\} \rightarrow (-,-,+,-,-,-,-,-,-,-) \\
& \{15\},\{2\},\{3\},\{4\} \rightarrow (-,-,-,+,-,-,-,-,-,-) \\
& \{23\},\{1\},\{4\},\{5\} \rightarrow (-,-,-,-,+,-,-,-,-,-) \\
& \{24\},\{1\},\{3\},\{5\} \rightarrow (-,-,-,-,-,+,-,-,-,-) \\
& \{25\},\{1\},\{3\},\{4\} \rightarrow (-,-,-,-,-,-,+,-,-,-) \\
& \{34\},\{1\},\{2\},\{5\} \rightarrow (-,-,-,-,-,-,-,+,-,-) \\
& \{35\},\{1\},\{2\},\{4\} \rightarrow (-,-,-,-,-,-,-,-,+,-) \\
& \{45\},\{1\},\{2\},\{3\} \rightarrow (-,-,-,-,-,-,-,-,-,+)
\end{array}
$$

In particular, vertices representing partitions of the same type consistently have similar behavior – they all have the same degree, and the number of elements of other partition types to which they are adjacent are consistent as well. Therefore we will break down these adjacencies according to partition type. The peak vertex is still solely adjacent to the 2-block partitions, so it will be omitted from further description here.

- **4-1 partitions:** As discussed for $L_{5,2}$ in Subsection 3.3.1, there are 5 of these vertices, each with 6 positive and 4 negative coordinates to indicate that 1 particular object out of 5 is separated from the rest. Each vertex has 44 neighbors, one of which is the peak. The neighborhood also includes all 14 additional 2-block partitions across both types, as well as all 3-block partitions across all types. Finally, each 4-1 partition vertex is neighbors with exactly 4 of the 2-1-1-1 vertices, and these neighbors are those whose pair includes the singleton from the 4-1 vertex and each one of the elements in the block of 4. This means the non-neighbors are exactly the 2-1-1-1 partitions which refine the 4-1 partition.

  Example 3.3.2. See Figure 3.3. The vertex representing $\{1234\}, \{5\}$ is neighbors with the vertices representing $\{15\}, \{2\}, \{3\}, \{4\}$; $\{25\}, \{1\}, \{3\}, \{4\}$; $\{35\}, \{1\}, \{2\}, \{4\}$; and $\{45\}, \{1\}, \{2\}, \{3\}$. In each of these, the element $\{5\}$ is part of the pair, whereas it is isolated in $\{1234\}, \{5\}$, meaning these partitions do not refine $\{1234\}, \{5\}$. This vertex is also neighbors with v_α and with all 2-block and 3-block partitions. This leaves exactly 6 vertices to which it is not adjacent, all among the 2-1-1-1 partitions, each of which is a refinement of the partition $\{1234\}, \{5\}$.

- **3-2 partitions:** Recall from $L_{5,2}$ in Subsection 3.3.1 that there are 10 of these vertices, each with 4 positive and 6 negative coordinates to represent the partition. Each vertex has 46 neighbors. As with the 4-1 partitions, each of these vertices is adjacent to the peak, to all other 2-block partition vertices, and to all 3-block partition vertices. However, each of these vertices is adjacent to 6 of the 2-1-1-1 vertices instead of 4. In particular, each 3-2 vertex is adjacent to

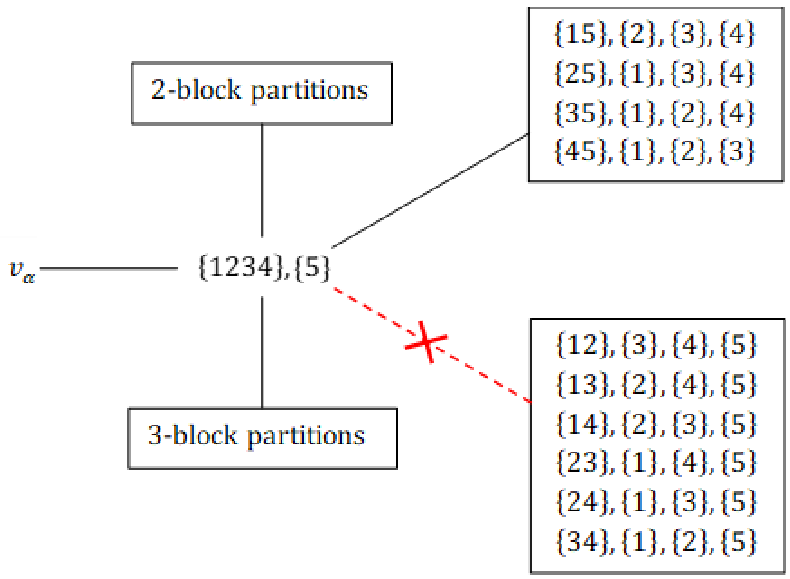


Figure 3.3: The 4-1 partition vertex $\{1234\}, \{5\}$ is not adjacent to any 2-1-1-1 block partition which refines it, but is adjacent to all other vertices. Adjacencies are indicated with solid black lines, non-adjacencies are indicated with dotted red lines.

the 2-1-1-1 vertices whose pair includes one element from the block of 3 and one element from the block of 2. This description closely mirrors the paradigm for the 4-1 vertices' neighbors among the 2-1-1-1 vertices. In both this case and the 4-1 partition case, the neighbors among the 2-1-1-1 partition vertices are explicitly those which do not represent refinements of the fixed 2-block partition.

Example 3.3.3. See Figure 3.4. The partition vertex $\{123\}, \{45\}$ is adjacent to v_α and to all other partitions with 2 or 3 blocks. Among the 4-block vertices, $\{123\}, \{45\}$ is adjacent solely to the vertices which do not refine it:

$$\{14\}, \{2\}, \{3\}, \{5\}$$

$$\{15\}, \{2\}, \{3\}, \{4\}$$

$$\{24\}, \{1\}, \{3\}, \{5\}$$

$$\{25\}, \{1\}, \{3\}, \{4\}$$

$$\{34\}, \{1\}, \{2\}, \{5\}$$

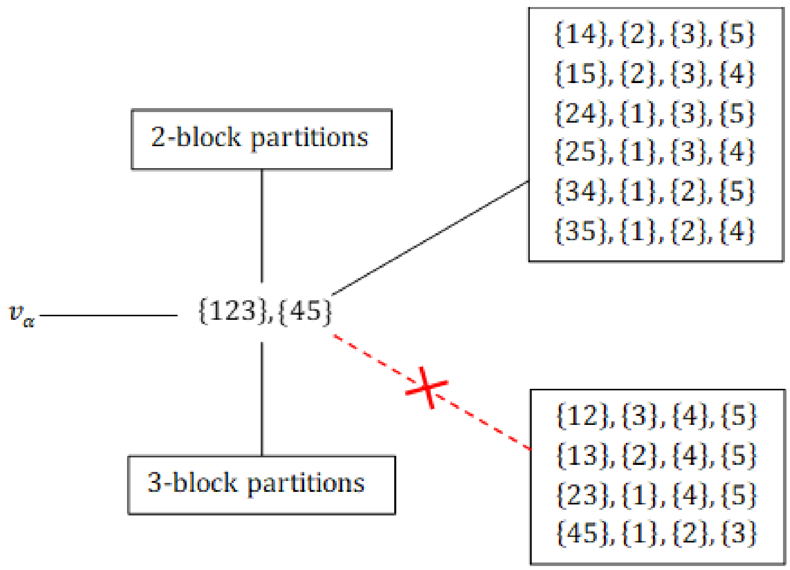


Figure 3.4: The 3-2 partition vertex representing $\{123\}, \{45\}$ is not adjacent to any 2-1-1-1 partition which refines $\{123\}, \{45\}$, but it is adjacent to all other vertices. Adjacencies are indicated with solid black lines, non-adjacencies are indicated with dotted red lines.

$$\{35\}, \{1\}, \{2\}, \{4\}$$

- **3-1-1 partitions:** In accordance with $L_{5,3}$ and Subsection 3.3.2, there are 10 of these vertices, each with 3 positive and 7 negative coordinates to represent the partition. Each has 45 neighbors in all. In alignment with the above descriptions, each of these vertices is adjacent to all 2-block partition vertices. Each of these is also adjacent to all other 3-1-1 partition vertices; however, in contrast to the $L_{5,3}$ case, a given 3-1-1 partition vertex is only adjacent to 12 of the 15 2-2-1 partition vertices. Finally, each of these vertices is adjacent to all but one of the 2-1-1-1 partition vertices.

Considering the 3 non-neighbors among the 2-2-1 partition vertices and the sole 2-1-1-1 non-neighbor, the common element is that the singleton elements from the 3-1-1 partition have been paired together as a block of 2. However,

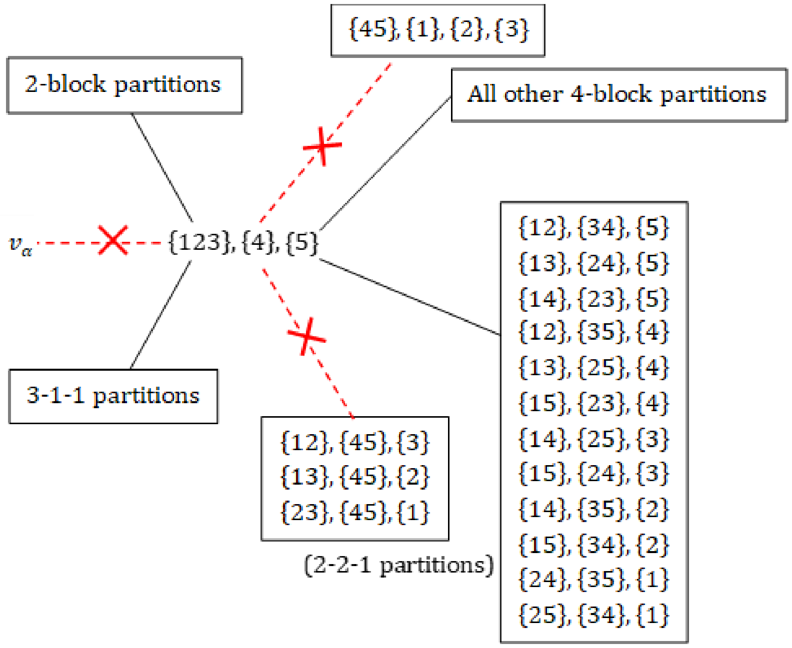


Figure 3.5: The 3-1-1 partition vertex $\{123\}, \{4\}, \{5\}$ is not adjacent to any 2-2-1 or 2-1-1-1 partition which has the block $\{45\}$, nor is it adjacent to v_α. It is adjacent to all other vertices. Adjacencies are indicated with solid black lines, non-adjacencies are indicated with dotted red lines.

this trend has no direct interpretation with regards to refinement, as no 2-2-1 partition can refine a 3-1-1 partition and the 2-1-1-1 neighbors of any fixed 3-1-1 partition include both refinements and non-refinements. An alternative explanation may stem from geometric restrictions.

Example 3.3.4. See Figure 3.5. The vertex $\{123\}, \{4\}, \{5\}$ is neighbors with all 2-2-1 partition vertices except $\{12\}, \{45\}, \{3\}$, $\{13\}, \{45\}, \{2\}$, and $\{23\}, \{45\}, \{1\}$; similarly, it is neighbors with all 2-1-1-1 partition vertices except $\{45\}, \{1\}, \{2\}, \{3\}$. These non-neighbors each contain the block $\{45\}$, whereas these two elements are in separate size-1 blocks in $\{123\}, \{4\}, \{5\}$. In addition, $\{123\}, \{4\}, \{5\}$ is adjacent to all other 3-1-1 partition vertices and all of the 2-block partition vertices. However, it is not adjacent to v_α.

- **2-2-1 partitions:** Per the discussion of $L_{5,3}$ in Subsection 3.3.2, there are 15 of these vertices, each with 2 positive and 8 negative coordinates. Each positive element indicates one of the independent blocks of 2 objects. Any 2-2-1 partition vertex has 43 neighbors. They are each adjacent to all 2-block partition vertices, as well as to all other 2-2-1 partition vertices. Each is additionally adjacent to 8 of the 10 3-1-1 partition vertices, but only 6 of the 10 2-1-1-1 partition vertices.

As with the 3-1-1 partitions, it is again most informative to examine which vertices are not adjacent to a particular 2-2-1 partition vertex, rather than those which are. For a given 2-2-1 partition vertex, the two non-neighbors among the 3-1-1 partition vertices are those whose group of 3 consists of the singleton element from the 2-2-1 plus one of the two pairs. As for the non-neighboring 2-1-1-1 partition vertices, they are precisely those whose pair contains the singleton from the 2-2-1 partition vertex.

Example 3.3.5. See Figure 3.6. The vertex representing the 2-2-1 partition $\{12\}, \{34\}, \{5\}$ is neighbors with all 3-1-1 partition vertices except $\{125\}, \{3\}, \{4\}$ and $\{345\}, \{1\}, \{2\}$, each of which has split one of the size-2 blocks from $\{12\}, \{34\}, \{5\}$ into a pair of size-1 blocks. Also, $\{12\}, \{34\}, \{5\}$ is not adjacent to the vertices representing the 4-block partitions $\{15\}, \{2\}, \{3\}, \{4\}$; $\{25\}, \{1\}, \{3\}, \{4\}$; $\{35\}, \{1\}, \{2\}, \{4\}$; and $\{45\}, \{1\}, \{2\}, \{3\}$. For each of these, the size-2 block contains the element $\{5\}$, which is the lone size-1 block in $\{12\}, \{34\}, \{5\}$. Beyond these exclusions, $\{12\}, \{34\}, \{5\}$ is adjacent to all other 2-block, 3-block, and 4-block partition vertices. It is not adjacent to v_α.

- **2-1-1-1 partitions:** Although most adjacencies have been covered by the previous descriptions, it is also valuable to consider them from the perspective of the new 2-1-1-1 partition vertices. There are 10 of these vertices, each with 1 positive coordinate indicating which two elements are together in the pair. Each has 35 neighbors in all, a set which includes all 9 other 2-1-1-1 partition vertices.

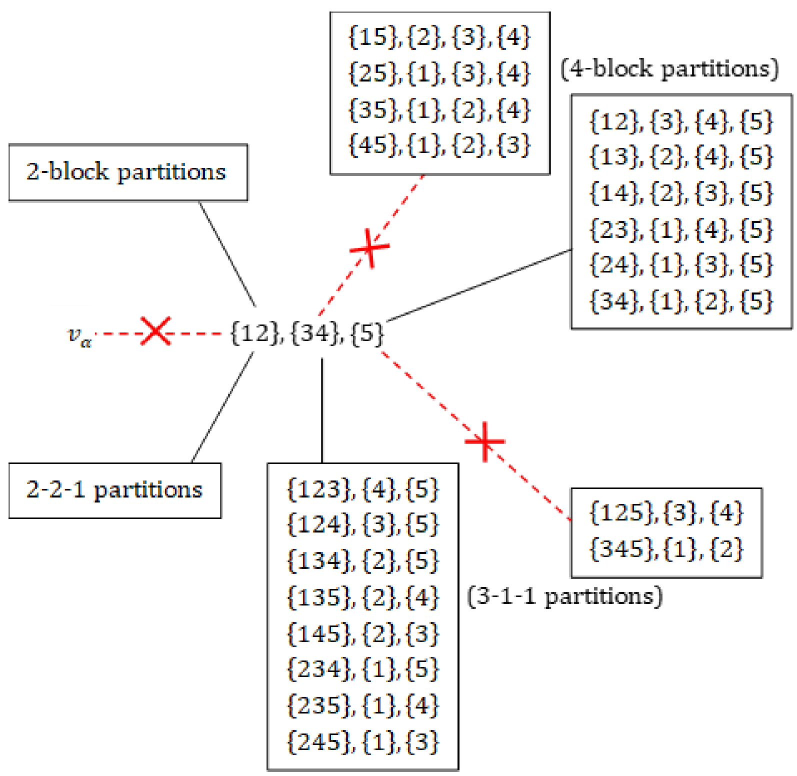


Figure 3.6: The 2-2-1 partition vertex $\{12\}, \{34\}, \{5\}$ is not adjacent to any 4-block partition that integrates $\{5\}$ into its size-2 block, to either 3-1-1 partition that splits $\{12\}$ or $\{34\}$ into a pair of size-1 blocks, or to v_α. Adjacencies are indicated with solid black lines, non-adjacencies are indicated with dotted red lines.

For both the 4-1 and 3-2 classes of partitions with 2 blocks, a given 2-1-1-1 partition vertex is neighbors with all vertices representing partitions in which the pair from the 2-1-1-1 has been split up. In other words, the *non-neighboring* 2-block partition vertices for any 2-1-1-1 partition are explicitly the coarsenings of that 2-1-1-1 partition. These relationships account for another 8 of the 35 neighbors, leaving 18 neighbors among the vertices representing 3-block partitions.

For a fixed 2-1-1-1 partition vertex, 9 out of 10 of the 3-1-1 partition vertices are its neighbors; the sole exception is the partition for which the pair from the 2-1-1-1 is comprised of the singletons from the 3-1-1. By contrast, a fixed 2-1-1-1 partition is adjacent to 9 out of 15 of the 2-2-1 partition vertices, meaning there are a total of 6 with which it is not neighbors. Among these non-neighboring 2-2-1 partition vertices, the common factor is that for each of them, the singleton is a member of the pair in the 2-1-1-1 partition vertex. For instance, $\{12, 3, 4, 5\}$ has the pair block [12] and is neighbors with $\{13, 24, 5\}$ even though this partition splits up the pair; however, $\{12, 3, 4, 5\}$ is not neighbors with $\{23, 45, 1\}$, in which the singleton is one of the elements of that pair.

Example 3.3.6. See Figure 3.7. The vertex representing $\{12\}, \{3\}, \{4\}, \{5\}$ is adjacent to all other 4-block partitions. It is not adjacent to v_α, nor to any 2-block partition that coarsens it. Of the 4-1 partitions, it is not adjacent to $\{1234\}, \{5\}$; $\{1235\}, \{4\}$; or $\{1245\}, \{3\}$. Of the 3-2 partitions, it is not adjacent to $\{123\}, \{45\}$; $\{124\}, \{35\}$; $\{125\}, \{34\}$; or $\{345\}, \{12\}$, which preserves the pair $\{12\}$ as its size-2 block. It is adjacent to all other 2-block partitions. In addition, $\{12\}, \{3\}, \{4\}, \{5\}$ is adjacent to all 3-1-1 partition vertices except $\{345\}, \{1\}, \{2\}$, which splits up the block $\{12\}$ and also brings together the three size-1 blocks $\{3\}, \{4\}, \{5\}$. Finally, among the 2-2-1 partitions, $\{12\}, \{3\}, \{4\}, \{5\}$ is adjacent to any partition that either retains the block $\{12\}$ or keeps both elements in blocks of size 2. However, it is not adjacent to any 2-2-1 partition which has either $\{1\}$ or $\{2\}$ as its size-1 block. Its non-neighbors in this category are:

$$\{13\}, \{45\}, \{2\}$$
$$\{14\}, \{35\}, \{2\}$$
$$\{15\}, \{34\}, \{2\}$$
$$\{23\}, \{45\}, \{1\}$$
$$\{24\}, \{35\}, \{1\}$$
$$\{25\}, \{34\}, \{1\}$$

3.3.4 $L_{5,5}$

In $L_{5,5}$, only one new vertex is added, for a total of 52. This is the basal vertex v_ω representing the partition of [5] into five blocks each containing a single object. For this polytope, a positive entry in the vertex sign pattern indicates a value of 1, and a negative entry indicates a value of $-\frac{1}{4}$.

$$v_\omega : \{1\}, \{2\}, \{3\}, \{4\}, \{5\} \to (-, -, -, -, -, -, -, -, -, -)$$

In comparison to $L_{5,4}$ above, very little is different about the neighborhoods of each vertex in $L_{5,5}$. The primary change to have occurred is that the 2-1-1-1 partition vertices are no longer all adjacent to one another, and that each of them is adjacent

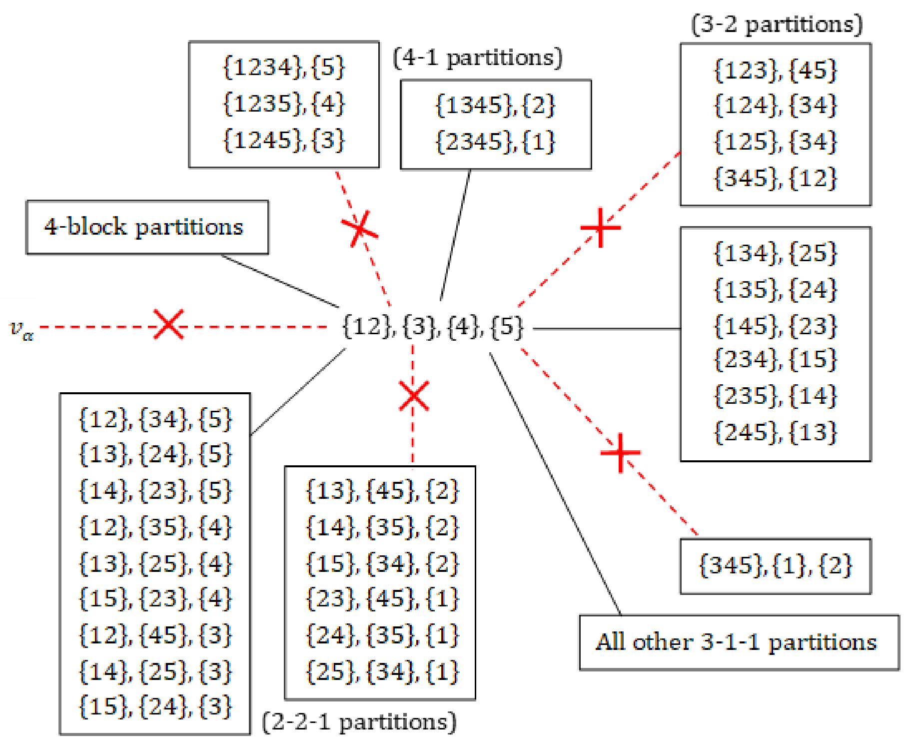


Figure 3.7: The 2-1-1-1 partition vertex $\{12\},\{3\},\{4\},\{5\}$ is not adjacent to any 2-block partition that coarsens it, or to v_α. It is adjacent to every 2-2-1 partition such that the size-1 block is not $\{1\}$ or $\{2\}$, and to all 3-1-1 partitions except $\{345\},\{1\},\{2\}$. It is also adjacent to all other 4-block partitions. Adjacencies are indicated with solid black lines, non-adjacencies are indicated with dotted red lines.

to the single 1-1-1-1-1 partition vertex. Specifically, each 2-1-1-1 partition vertex is now adjacent only to the 2-1-1-1 partition vertices whose pair blocks share an element with it.

Example 3.3.7. The vertex representing the 2-1-1-1 partition vertex $\{12\}, \{3\}, \{4\}, \{5\}$ is now adjacent to v_ω, but it is no longer adjacent to the 4-block partitions $\{34\}, \{1\}, \{2\}, \{5\}$; $\{35\}, \{1\}, \{2\}, \{4\}$; and $\{45\}, \{1\}, \{2\}, \{3\}$. All other adjacencies and non-adjacencies of this partition described for $L_{5,4}$ and shown in Figure 3.7 remain consistent.

Meanwhile the basal vertex is adjacent to all of the 2-1-1-1 partition vertices, but not to any other vertices. The neighborhoods of v_α, the 2-block vertices, and the 3-block vertices are identical to those described in Subsection 3.3.3 for $L_{5,4}$.

3.3.5 Iterative faces

An interesting phenomenon occurs among the middle-dimensional faces of $L_{5,k}$. For each value of k, we can more closely consider the 6-dimensional faces, in search of interesting structures that might be related to the $L_{4,k}$ family. Through this line of investigation, we find that among the 5060 six-dimensional faces of $L_{5,2}$, there are precisely 20 faces containing exactly 8 vertices, which is the number of vertices possessed by $L_{4,2}$. All other 6-dimensional faces of $L_{5,2}$ include only 7 vertices. By listing out the vertices in each of these 20 unusual faces, we can seen that each such face has a unique rule obeyed by all of its vertices, and that these rules pertain to the set of 5 objects whose partitions are represented by the vertices.

10 of the faces have rules that glue a pair of objects together and treat them as a single entity. For example, one face includes the exact set of vertices in which the

objects $\{1\} \in [5]$ and $\{2\} \in [5]$ are in the same block of the corresponding partition. The partitions in question are the following eight arrangements:

$$\begin{aligned}
\text{One 1-block partition: } & \{12345\}; \\
\text{Three 4-1 partitions: } & \{1234\}, \{5\}; \{1235\}, \{4\}; \{1245\}, \{3\}; \\
\text{Four 3-2 partitions: } & \{123\}, \{45\}; \{124\}, \{35\}; \{125\}, \{34\}; \{345\}, \{12\}
\end{aligned}$$

Each possible pair i, j that can be selected from the set of five objects has a corresponding face according to this paradigm, for a total of 10 faces. The remaining 10 faces of interest also each represent a fixed relationship between a specific pair of objects, but instead of gluing the two together, the pair is forced apart. In the separation case, each face contains the exact set of vertices in which its defining pair of elements are in different blocks of the associated partitions. For example, considering $\{1\} \in [5]$ and $\{2\} \in [5]$ again, there is a 6-dimensional face incident exactly on the 8 vertices representing the following partitions:

$$\begin{aligned}
\text{Two 4-1 partitions: } & \{1345\}, \{2\}; \{2345\}, \{1\} \\
\text{Six 3-2 partitions: } & \{134\}, \{25\}; \{135\}, \{24\}; \{145\}, \{23\} \\
& \{234\}, \{15\}; \{235\}, \{14\}; \{245\}, \{13\}
\end{aligned}$$

Further investigation of the $k = 3$ and $k = 4$ cases reveals that the first set of unusual 6-dimensional faces – those representing sets of partitions in which a specific pair of objects was fixed together – start a longer trend. In both $L_{5,3}$ and $L_{5,4}$, a unique set of ten 6-dimensional faces exists that contain the exact and complete set of vertices within $L_{5,k}$ in which each possible pair of objects is held together throughout all partitions in their face. These faces also continue to represent the highest vertex incidence counts among all 6-dimensional faces in $L_{5,k}$.

3.4 Analysis

From the observations made throughout this section, several trends emerge that motivate further inquiry into broad behavior. To begin with, in both the $n = 4$ and $n = 5$ cases, the peak vertex is consistently adjacent only to the 2-block partition vertices, regardless of the value of k. At the other extreme, although the basal vertex only appears in the $k = n$ polytope, it is adjacent solely to the $(n - 1)$-block partitions for both $n = 4$ and $n = 5$. In the next chapter, Proposition 4.1.1 and Proposition 4.1.3 will show that these statements hold for all polytopes $L_{n,k}$.

Next, the $n = 5$ observations for $k = 4$ suggest that refinement relationships between partitions play some role in edge structure; however, the exact nature of this role is unclear from the concrete cases, and there may be additional mechanisms at play that complicate the overall picture. In particular, although the partition lattice on $n = 4$ is contained in the edge set of $L_{4,4}$ as shown in Figure 3.2, the polytope has far more edges than are found in the lattice. The same is true for $L_{5,5}$ in comparison to the $n = 5$ partition lattice, indicating that there must be much more governing the existence of edges. Conjecture 4.1.6 and Proposition 4.1.8 address the uncertainty around this observation and a specific case that can be fully proved.

One subset of edges that require more consideration in the context of this refinement observation are the edges from 2-block partitions to 4-block partitions, which appear to be anticorrelated with refinement relationships. There may be a geometric reason why these refinement relationships do not produce edges – perhaps, for example, the intervening 3-block partitions in the lattice portion of the edge set make an edge geometrically impossible. Further investigation is needed to determine the significance of these edges.

In addition, for all $n = 4$ and $n = 5$ cases, it can be observed that regardless of the value of k, all k-block partitions are adjacent to each other, e.g. all 3-block partitions are adjacent to each other in $L_{5,3}$. This is especially interesting since there are multiple categories of 3-block partitions for $n = 5$ and the associated vertices have different characteristics. A result for $k = n - 1$ is obtained explicitly in Proposition 4.1.9 and extended for $(n-1)$-block partitions in $L_{n,n}$ by Proposition 4.1.10. More information is available for $k = 2$, but other k-values remain open.

Finally, the iterative faces observed in the $n = 5$ polytopes are highly compelling in regards to consistent structure throughout this class of polytopes. As mentioned earlier, although $L_{5,2}$ has many nice symmetries and behaviors, a large portion of these are lost as k increases. However, the potential presence of lower-dimensional faces that belong to the same family of polytopes suggests that there may be a much more regularized internal structure than previously known. Within the case studies, these structures could explain the presence and absence of additional edges whose patterns cannot be discerned by examining the edge behavior alone if it can be proved that they are in fact combinatorially equivalent to the lower-dimensional polytopes. In the next chapter, the existence and desired equivalence of these faces is proved for general n, k with Lemma 4.2.1, Theorem 4.2.4, and Corollary 4.2.5.

Chapter Four

Geometric results

The observations made in Chapter 3 using models of the $n = 4$ and $n = 5$ polytopes resulted in a number of conjectures. The conjectures center primarily on edge structure for the up-to-k-partition polytopes, but also investigate the possibility of $L_{n,k}$ containing lower-dimensional partition polytopes as faces. In this chapter, we prove some of these statements for general n, k to establish a theoretical understanding of the low-dimensional facial structure of $L_{n,k}$, and determine which questions remain open. Throughout the chapter, we will use $d = \binom{n}{2}$ to represent the dimension of $L_{n,k}$. Unless otherwise specified, it is assumed that $L_{n,k}$ resides in $\mathbb{R}^d$.

4.1 Edge Structure

We begin with several results regarding the edges of the up-to-k partition polytopes. These proofs establish the consistent presence and logic of several subsets of edges. We first consider the edges incident on the peak and basal vertices. Recall from Definition 3.0.1 that the peak vertex v_α is the point $(1, ..., 1) \in \mathbb{R}^d$. Lemma 2.3.5, which was proved in [6], states that in $L_{n,2}$, every vertex is adjacent to every other vertex. We will need this fact in order to prove the first result.

Proposition 4.1.1. *For any $n \geq 3, 2 \leq k \leq n$, the peak vertex is exactly adjacent to the 2-block partition vertices.*

Proof. From [6], the claim is known to be true for $k = 2$. Consider any 2-block partition vertex v_1 in $L_{n,2}$; this point will be a collection of $\{1, -1\}$, and therefore the line segment from v_α to v_1 will maintain some coordinates at 1 and will smoothly vary the rest from 1 to -1, such that any point along the line with have some corresponding value $-1 \leq \lambda \leq 1$ for each coordinate that is -1 in v_1. Let v_2 be the vertex in $L_{n,k}$ that represents the same 2-block partition as v_1; then v_2 has the same sign pattern as v_1 but with $-\frac{1}{k-1}$ in place of each -1, meaning it lies on the line between v_α and v_1. Note that according to [34], the collection of facet-defining inequalities whose facets are incident on v_α in $L_{n,2}$ are also facet-defining for any $L_{n,k}$. Since the line between v_α and v_1 is an edge in $L_{n,2}$, it must lie in the intersection of $d-1$ of these facets; as the inequalities defining the facets are preserved with the increase in k, this line is still in the intersection of $d-1$ of these facets, meaning it must still be an edge. Therefore v_2 is adjacent to v_α. Moreover, any vertex representing a partition with more than 2 blocks can be represented as a linear combination of v_α with some subset of the 2-block partition vertices using positive coefficients, meaning that no additional facets created by adding these vertices can contain v_α. As such, v_α cannot be adjacent to any such vertices. $\square$

From Proposition 4.1.1 we can observe that the peak vertex has $2^{n-1}-1$ edges incident on it, regardless of n and k. It is interesting to note that there is a clear interpretation of these edges in terms of Gram matrices and the corresponding vector configurations. Consider any 2-block partition; we can think of the two blocks as indicating two sets of vectors that are each fully aligned with each other, and if moved will move in synchrony. For $k=2$, a 2-block partition vertex will have entries of -1 to indicate pairs occupying different blocks, which in the Gram matrix setting corresponds to vectors pointing in opposite directions. Any point along the edge from the 2-block vertex to v_α will still have all the same $+1$ entries as the 2-block vertex, meaning that the corresponding vector configuration will still have the same sets of vectors glued together as the 2-block partition. However, the other entries in such a point will all be equal to some other value α, which can be seen as the inner product of two unaligned unit vectors. Therefore, the vector configuration at any point along the edge will be one in which the two groups of synchronized unit vectors are still present, but are neither aligned with each other nor directly opposing each other.

We now turn to the basal vertex v_ω, which exists solely in $L_{n,n}$. To obtain a result for the adjacencies of v_ω, we must first characterize a subset of facets of $L_{n,n}$.

Lemma 4.1.2. *For any $n \geq 3$, the up-to-n partition polytope $L_{n,n}$ has d facets defined by the inequalities $x_i \geq -\frac{1}{n-1}, 1 \leq i \leq d$.*

Proof. By definition of the vertex set of $L_{n,n}$, the inequality $x_i \geq -\frac{1}{n-1}$ is valid for each $1 \leq i \leq d$ Note that for any n, there are exactly $d = \binom{n}{2}$ $(n-1)$-block partition vertices, which each specify a partition with a single block of size 2 and all other $n-2$ blocks containing a single object apiece. In $L_{n,n}$, each $(n-1)$-block partition vertex v will have $v_i = 1$ for exactly one $i \in [d]$ and $v_j = -\frac{1}{n-1}$ otherwise, which means that the pair of objects represented by i form the size-2 block and all other pairs of objects are in different blocks. Moreover, each of the $\binom{n}{2}$ $(n-1)$-block partition vertexs will have their singular positive coordinate $v_i = 1$ at a different value of i. Therefore, for each hyperplane $x_i = -\frac{1}{n-1}$, $d-1$ of the d vertices for $(n-1)$-block partitions will lie on the hyperplane.

Furthermore, the $d-1$ vertices described above in conjunction with the basal vertex v_ω comprise an affine independent set of points. We can observe that the larger collection of points consisting of all d $(n-1)$-block partition vertices in $L_{n,n}$ as well as the basal vertex is affine independent. Let v^i be the $(n-1)$-block partition vertex such that the i-th entry of v^i is 1, and all other entries are $-\frac{1}{n-1}$. Then:

$$(v^i - v_\omega)_j = \begin{cases} 1 + \frac{1}{n-1}, & j = i \\ 0, & j \neq i \end{cases}$$

In other words, the collection of points resulting from these subtractions each sit on a unique axis, making them linearly independent. Therefore the collection of points v_ω, v^i over $1 \leq i \leq d$ is a set of $d+1$ affine independent points.

Each hyperplane $x_i = -\frac{1}{n-1}$, contains $d-1$ of the $(n-1)$-block partition vertices, along with the v_ω, for a total of d points from the affine independent set identified above. Thus we have d affine independent points on each hyperplane $x_i = -\frac{1}{n-1}$. These points are all also vertices of $L_{n,n}$. As such, each valid inequality $x_i \geq -\frac{1}{n-1}$ must give a $(d-1)$-dimensional face of $L_{n,n}$, meaning the polytope has d such facets in total. □

Lemma 4.1.2 establishes that $L_{n,n}$ will have a collection of facets given by the d inequalities $x_i \geq -\frac{1}{n-1}, 1 \leq i \leq d$. With this knowledge, we can now focus on the basal vertex, which is unusual in that it sits on every single one of those facets. This allows us to achieve a key result regarding v_ω.

Proposition 4.1.3. *For $L_{n,n}$, the basal vertex is adjacent to the vertices representing the $(n-1)$-block partitions, and is not adjacent to any other vertices.*

Proof. By definition, the basal vertex v_ω has the form $(-\frac{1}{n-1}, ..., -\frac{1}{n-1})$ regardless of n. In addition we can observe that v_ω sits at the intersection of the d hyperplanes that mark the boundaries of the halfspaces $x_i \geq -\frac{1}{n-1}$ which define $L_{n,n}$. As shown in Lemma 4.1.2, these hyperplanes must be among the defining hyperplanes of $L_{n,n}$. Similarly, the vertex for any $(n-1)$-partition will possess exactly one 1 in its coordinate vector, and all other entries will be $-\frac{1}{n-1}$. This means any such vertex sits at the intersection of $d-1$ of the $x_i \geq -\frac{1}{n-1}$ facets.

From this structure, we can see that every point on the line between v_ω and an $(n-1)$-block partition vertex must have the value $-\frac{1}{n-1}$ at all coordinates except one, which is smoothly varying between $\frac{1}{n-1}$ and 1. Therefore the line itself also lies in the intersection of $d-1$ of the $x_i \geq -\frac{1}{n-1}$ inequalities, and thus is the intersection of $d-1$ defining hyperplanes of $L_{n,n}$. This line must therefore be an edge, so any $(n-1)$-partition must be adjacent to v_ω.

We also note that by construction, v_ω is exactly the point of intersection of the d facets from the $x_i \geq -\frac{1}{n-1}$ inequalities, meaning these must be the sole facet-defining hyperplanes on which v_ω sits; therefore v_ω cannot be adjacent to any other vertices, as these exist on at most $d-2$ of these hyperplanes. □

Put together, there is a nice symmetry to Proposition 4.1.1 and Proposition 4.1.3. At both the "top" and "bottom" of the polytope, we have adjacency exclusively with the proximal level of partitions. These portions of the edge structure will thus align exactly with the partition lattice for any n. Unfortunately, as seen in the case studies, plenty of edges can exist in $L_{n,k}$ outside the constraints of the partition lattice. However, we can use this intuition to explore other edges based on partition relationships, particularly refinement.

4.1.1 Refinement and Associated Edges

To begin inquiries pertaining to refinement, we must establish some terminology.

Definition 4.1.4. Let P be a partition on n objects, with m non-empty blocks. We call a partition P' a **direct refinement** of P, and say it **directly refines** P, if P' is a refinement of P with exactly $m+1$ non-empty blocks.

Direct refinement represents the covering relationship of the partition lattice; by this definition, the statement "P covers P' in the partition lattice" is equivalent to the statement "P' directly refines P". Note that to achieve the criterion that a direct refinement has $m+1$ non-empty blocks, it must be true that $m-1$ of the blocks of P are blocks of P', and that the remaining 2 blocks of P' are a 2-block partition of the remaining 1 block of P. In other words, P' is constructed by splitting a single block of P into two non-empty blocks, and leaving all other blocks of P as they are.

Example 4.1.5. Consider the partition $\{123, 45\}$ of $[5]$. There are exactly four direct refinements of this partition: three which refine the block $\{123\}$, and one which refines the block $\{45\}$. The four direct refinements are:

$$\begin{aligned} \text{2-2-1 partitions: } & \{12\}, \{45\}, \{3\} \\ & \{13\}, \{45\}, \{2\} \\ & \{23\}, \{45\}, \{1\} \\ \text{3-1-1 partition: } & \{123\}, \{4\}, \{5\} \end{aligned}$$

In the case studies, we observed that the full partition lattice for $n = 4$ is contained within the edge structure of $L_{4,4}$, and likewise for $n = 5$; additionally, for $k < n$, the edge structures contained the portion of the partition lattice extending from the 1-block partition at the top down to the tier of k-block partitions. Based on these observations, we have the following conjecture:

Conjecture 4.1.6. *Consider $L_{n,k}$ and let P, P' be two partitions on n elements such that P' directly refines P and P' has at most k non-empty blocks. Then $L_{n,k}$ has an edge between the vertices $v_P, v_{P'}$ representing these partitions.*

However, Conjecture 4.1.6 has proved difficult to verify in generality. A special case is when P' is obtained specifically by splitting a block of size 2 in P. To prove the result in this case, we first establish a useful lemma pertaining to normal cones of vertices of $L_{n,k}$.

Lemma 4.1.7. *Any vertex v of $L_{n,k}$ has a sign pattern unique among the vertices of $L_{n,k}$, and the orthant of $\mathbb{R}^d$ corresponding to this sign pattern is contained in the normal cone of v.*

Proof. Note that v represents a unique partition of $[n]$. Each partition is represented using values of $\{1, -\frac{1}{k-1}\}$ at each coordinate, with a positive value indicating a pair of objects in $[n]$ that are in the same block of the partition represented by v, and a negative value indicating a pair of objects that are in different blocks. Thus regardless of k, the sign pattern of v remains constant, and since the sign pattern is determined by the partition, it is unique to v; no other vertices of $L_{n,k}$ can have the same sign pattern as v.

Let $x \in \mathbb{R}^d$ have the same sign pattern as v. For x to be in the normal cone of v, it must be true that $x \cdot v \geq x \cdot w$ for all other vertices w of $L_{n,k}$. Note that for all $1 \leq i \leq d$, we have $x_i v_i \geq 0$ since x_i and v_i will have the same sign. Moreover, the coordinate values for the vertices ensure that for any $1 \leq i \leq d$ and any vertex $w \neq v$, we will either have $x_i w_i = x_i v_i$ or $x_i w_i \leq 0$. Therefore, $x \cdot v \geq x \cdot w$. Thus x belongs to the normal cone of v, and correspondingly the full orthant of $\mathbb{R}^d$ which shares the sign pattern of v must belong to the normal cone of v. $\square$

With Lemma 4.1.7 showing that each vertex of $L_{n,k}$ contains the orthant with its sign pattern in its normal cone, we can obtain a result on direct refinement of blocks containing two elements.

Proposition 4.1.8. *Consider $L_{n,k}$ and let P, P' be two partitions on n elements such that P' directly refines P, P' has at most k non-empty blocks, and the block of P that is refined to produce P' contains exactly two elements. Then $L_{n,k}$ has an edge between the vertices $v_P, v_{P'}$ representing these partitions.*

Proof. The line between the two partition vertices $v_P, v_{P'}$ is an edge of the polytope if and only if its normal cone (Definition 2.1.8) is $(d-1)$-dimensional. Note that as discussed in Section 2.1, the normal cone of this line is exactly the intersection of the normal cones of its two endpoint vertices. Let A, B be the unique blocks in P' that resulted from the refinement; then $A \cup B$ is a block of P, and $|A| = |B| = 1$. Recall that by the structure of vertices as previously described, a fixed vertex has a sign pattern that is unique among the vertices of $L_{n,k}$, where each element of that sign pattern indicates whether a specific pair of objects are in the same block of the corresponding partition. Thus v_P and $v_{P'}$ both have an identifying sign pattern. Moreover, by Lemma 4.1.7, for any fixed vertex, the orthant of $\mathbb{R}^d$ corresponding to its sign pattern must be contained in the normal cone of that vertex.

Since P' is a direct refinement of P, splitting one particular block into exactly two blocks A and B, the differences in the sign patterns of v_P and $v_{P'}$ will only be at the positions representing relationships between objects in A and objects in B. At v_P, these positions will be positive since blocks A and B are united; at $v_{P'}$, they will be negative. All other elements of the sign patterns, however, will be identical in v_P and $v_{P'}$ because the relationships of objects in A or B to objects outside A and B have been preserved. Since we are specifically concerned with the case where both A and B contain a single element, there are no relationships of objects within A or within B

to each other to worry about, but these would also be preserved. This all means that the sign patterns of v_P and $v_{P'}$ differ at precisely one location, representing whether the single object in A and the single object in B are in the same partition block.

We know that the orthants corresponding to the sign patterns of v_P and $v_{P'}$ are contained in their respective normal cones; in our case, these orthants intersect at a $(d-1)$-dimensional subspace of $\mathbb{R}^d$ where the signs of $d-1$ coordinates are specified and the final coordinate is restricted to 0. Therefore this intersection must be contained in the normal cone of the line between v_P and $v_{P'}$, which means the normal cone of that line has dimension at least $d-1$. Since the normal cone of any line in a polytope can have dimension at most $d-1$, it follows that this line has a normal cone of dimension exactly $d-1$, which means it is an edge of the polytope. $\square$

The structure of the proof of Proposition 4.1.8 gives some insight into what will be needed to prove the broader result. In particular, the goal is once again to show that the normal cone of the line between v_P and $v_{P'}$ is exactly $d-1$ dimensions, and that this cone is equal to the intersection of the normal cones of v_P and $v_{P'}$. However, now that the sizes of A and B can be larger than 1, the shared sign patterns of v_P and $v_{P'}$ will only produce a shared $(d-|A|\cdot|B|)$-dimensional space. This leaves $|A|\cdot|B|-1$ dimensions unaccounted-for amid the coordinates that change sign between v_P and $v_{P'}$. Note that to be in the normal cone of a vertex v, a point g must satisfy $g^T v \geq g^T w$ for all other vertices w in $L_{n,k}$. In this case, it must be true that $g^T v_P = g^T v_{P'} \geq g^T w$. Let I be the set of coordinates that change sign between v_P and $v_{P'}$; then a necessary condition to obtain $g^T v_P = g^T v_{P'}$ will be $\sum_{i\in I} g_i = 0$. This condition defines an affine space of $|A|\cdot|B|-1$ dimensions, but is not sufficient to guarantee that g will belong to the normal cone. Only once the additional necessary conditions are identified can the result be fully obtained.

4.1.2 Edges Within Partition Levels

So far, the focus has been on edges bridging partitions with different numbers of blocks; however, in the case studies it was shown that many edges exist between partitions with the same number of blocks. The next two propositions discuss a specific subset of these edges: the edges between $(n-1)$-block partition vertices in the $k = n-1$ and $k = n$ cases.

Proposition 4.1.9. *Let u, v be any two $(n-1)$-block partition vertices in $L_{n,n-1}$. Then an edge exists between u and v.*

Proof. Since u, v represent different partitions of n objects into $n-1$ blocks, they each consist of a single block of size 2 and all remaining blocks of size 1, and their blocks of size 2 will not be the same. In terms of coordinates, the partitions represented by u, v will be such that there exists some $i, j \in [d], i \neq j$ such that $u_i = v_j = 1$. Moreover, $u_\alpha = -\frac{1}{n-2}$ for $\alpha \neq i$, and $v_\alpha = -\frac{1}{n-2}$ for $\alpha \neq j$.

To show that an edge exists between these vertices, we will show that there exists an $(n-1)$-dimensional subset of $\mathbb{R}^d$ that is contained in the normal cones of both u and v (i.e. in the intersection of their normal cones). We first note that for any vector g to be in the normal cones of both u and v, it must be true that $g^T u = g^T v \geq g^T w$

for all other vertices $w \neq u, v$ of $L_{n,n-1}$. By the structure of u and v, we can see that:

$$
\begin{aligned}
g^T u &= g_i + \sum_{\alpha \neq i} g_\alpha \cdot \left(-\frac{1}{n-2}\right) = g_i - \frac{1}{n-2} \sum_{\alpha \neq i} g_\alpha \\
g^T v &= g_j + \sum_{\alpha \neq j} g_\alpha \cdot \left(-\frac{1}{n-2}\right) = g_j - \frac{1}{n-2} \sum_{\alpha \neq j} g_\alpha
\end{aligned}
$$

Therefore, to get $g^T u = g^T v$, we need:

$$
\begin{aligned}
g_i - \frac{1}{n-2} g_j &= g_j - \frac{1}{n-2} g_i \\
(1 + \frac{1}{n-2}) g_i &= (1 + \frac{1}{n-2}) g_j \\
g_i &= g_j
\end{aligned}
$$

We conclude that $g_i = g_j$ is a necessary condition for g to be in both normal cones, which means that $NC(u) \cap NC(v)$ can have dimension at most $d-1$. Moving forward, we fix $g_i = g_j = a$ for some constant value $a \in \mathbb{R}$.

Consider $G \subset \mathbb{R}^d$ defined as:

$$
G = \{g \in \mathbb{R}^d : g_\alpha < 0 \quad \forall \alpha \in [d], \text{ and } g_i = g_j, |g_i| \leq |g_\alpha| \quad \forall \alpha \neq i, j\}
$$

Note that under this definition we have $g_i = g_j = a < 0$. Therefore G is a $(d-1)$-dimensional cone in $\mathbb{R}^d$, and for any $g \in G$, we have:

$$
g^T u = g^T v = a \left(1 - \frac{1}{n-2}\right) - \frac{1}{n-2} \sum_{\alpha \neq i,j} g_\alpha
$$

We will show that this cone is contained in the intersection of normal cones of u and

v. This requires considering the other vertices of $L_{n,n-1}$ under a few different cases.

Case 1: Let $w \neq u, v$ be a $(n-1)$-block vertex of $L_{n,n-1}$. Then there exists some $\beta \in [d]$ such that $\beta \neq i, j$ and $w_\beta = 1$, whereas $w_\alpha = -\frac{1}{n-2}$ for all $\alpha \neq \beta$. For any $g \in G$, we compute $g^T w$ as:

$$\begin{aligned} g^T w &= g_\beta + \sum_{\alpha \neq \beta} g_\alpha \cdot \left(-\frac{1}{n-2}\right) \\ &= g_\beta + a \cdot \left(-\frac{2}{n-2}\right) - \frac{1}{n-2} \sum_{\alpha \neq \beta, i, j} g_\alpha \end{aligned}$$

We must show $g^T u \geq g^T w$, or in other words, $g^T u - g^T w \geq 0$. Note that $g_\alpha \leq 0$ for all coordinates $\alpha \in [d]$, including i, j at which $g_i = g_j = a$.

$$\begin{aligned} g^T u - g^T w &= \left(a(1 - \frac{1}{n-2}) - \frac{1}{n-2} g_\beta\right) - \left(g_\beta - \frac{2}{n-2} a\right) \\ &= a\left(1 + \frac{1}{n-2}\right) - g_\beta \left(1 + \frac{1}{n-2}\right) \end{aligned}$$

Since $1 + \frac{1}{n-2} > 0$ and since $|a| \leq |g_\beta|$ with $a, g_\beta < 0$ by the definition of G, we are guaranteed that $g^T u - g^T w \geq 0$, and thus $g^T u \geq g^T w$.

Case 2: Now let w be a vertex representing a partition with fewer than $n-1$ blocks. This means w has at least two 1s at some pair of locations. Note that if we were to break $g^T w$ up according to the positive and negative components of the sum, any component $g_\alpha w_\alpha$ will be positive if and only if $w_\alpha = -\frac{1}{n-2}$ since $g_\alpha < 0$. This means for any vertex w with **exactly** two 1s at some two locations, and another hypothetical vertex w' with those same two 1s and at least one additional 1 at some third location, $g^T w \geq g^T w'$. Therefore we can restrict this case to the situation where w has exactly two 1s at some locations $\beta, \gamma \in [d]$, in which case we will have:

$$g^T w = g_\beta + g_\gamma - \frac{1}{n-2} \sum_{\alpha \neq \beta,\gamma} g_\alpha$$

There are a few sub-cases here depending on the locations of those 1s.

2a: Suppose $w_i = w_j = 1$ and $w_\alpha = -\frac{1}{n-2}$ for all $\alpha \neq i, j$, i.e. w has its 1s at the same locations as u and v. Then when we compute $g^T u - g^T w$, we find:

$$\begin{aligned} g^T u - g^T w &= a\left(1 - \frac{1}{n-2}\right) - 2a \\ &= a\left(-1 - \frac{1}{n-2}\right) \end{aligned}$$

But since $a < 0$, this immediately yields $g^T u - g^T w \geq 0$ as desired.

2b: Suppose w has a 1 at exactly **one** of i or j; without loss of generality, we say w has a 1 at i and a 1 at some other location $\beta \neq j$. Now when we take $g^T u - g^T w$, we get:

$$g^T u - g^T w = -\frac{1}{n-2} g_\beta - g_\beta$$

But once again we have $g_\beta < 0$, which means this also gives $g^T u - g^T w \geq 0$.

2c: Finally, we consider the case where β, γ are both different from i and j, i.e. w has its two 1s at two different locations from u and v. Now computing our difference gives us:

$$\begin{aligned} g^T u - g^T w &= \left(a(1 - \frac{1}{n-2}) - \frac{1}{n-2}(g_\beta + g_\gamma)\right) - \left(-\frac{2}{n-2}a + g_\beta + g_\gamma\right) \\ &= (1 + \frac{1}{n-2})a - (1 + \frac{1}{n-2})(g_\beta + g\gamma) \end{aligned}$$

To get our desired outcome out of this, we need $a - (g_\beta + g_\gamma) \geq 0$, or $a \geq g_\beta + g_\gamma$. Since $a, g_\beta, g_\gamma < 0$, this is equivalent to requiring $|a| \leq |g_\beta + g_\gamma| = |g_\beta| + |g_\gamma|$. But it is a pre-existing condition of G that $|a| \leq |g_\alpha|$ for all $\alpha \neq i, j$, which guarantees that $|a| \leq |g_\beta| + |g_\gamma|$ regardless of which indices and values these yield. Therefore our desired outcome is guaranteed by the structure of G, and we have $g^T u - g^T w \geq 0$ once again.

After working through these two cases, we have that for any $g \in G$, $g^T u = g^T v \geq g^T w$ for any vertex $w \neq u, v$ of $L_{n,n-1}$. Therefore we conclude that the $(d-1)$-dimensional cone G is contained in both the normal cone of u and the normal cone of v, which means $NC(u) \cap NC(v)$ has dimension at least $d-1$. This means the intersection of the normal cones must have dimension exactly $d-1$, and therefore an edge exists between u and v. □

Next, we can prove a similar but distinct result for $(n-1)$-block partition vertices in $L_{n,n}$. Much of the argument structure is the same, but some key details have shifted due to the presence of the n-block vertex.

Proposition 4.1.10. *Let u, v be any two $(n-1)$-block partition vertices in $L_{n,n}$. Then an edge exists between u and v if and only if the unique size-2 blocks of the partitions corresponding to u and v have a non-empty intersection.*

Proof. We note that the structures of u, v in this case are identical to their structures in the previous proposition, but with $-\frac{1}{n-1}$ in place of $-\frac{1}{n-2}$ at all but one coordinate location: there exist some $i, j \in [d], i \neq j$ such that $u_i = v_j = 1$. Therefore, the initial necessary condition for a vector $g \in \mathbb{R}^d$ to be in the normal cones of both vertices is still present. By the same reasoning as the previous proof, we must have $g_i = g_j = a$ for some constant value a in order to get $g^T u = g^T v$, which is required in order to have $g \in NC(u) \cap NC(v)$. As such, we already know that this intersection can once again have dimension at most $d-1$. We will first show that this intersection cannot have dimension $d-1$ if the size-2 blocks of u and v are disjoint. Define:

$$G_0 = \{g \in \mathbb{R}^d : g_i = g_j\}$$

Assume that u and v represent $(n-1)$-block partitions with **disjoint** size-2 blocks, meaning that i and j such that $u_i = v_j = 1$ represent non-overlapping pairs of objects (out of the n objects). For any vector $g \in G_0$, we have:

$$g^T u = g^T v = a(1 - \frac{1}{n-1}) + \sum_{\alpha \neq i,j} g_\alpha \left(-\frac{1}{n-1}\right)$$

For any such u, v, there exists a unique vertex w^* which has $w_i = w_j = 1$ and $w_\alpha = -\frac{1}{n-1}$ for all $\alpha \neq i, j$. This vertex represents the most refined mutual coarsening of the partitions for u and v, which is a partition containing exactly two size-2 blocks matching those of u and v and all other blocks being singletons (so $n-2$ blocks in all). Given any $g \in G_0$, this vertex will have:

$$g^T w^* = 2a + \sum_{\alpha \neq i,j} g_\alpha \left(-\frac{1}{n-1}\right)$$

Observe that:

$$g^T u - g^T w^* = a(1 - \frac{1}{n-1}) - 2a = -\frac{n-2}{n-1} a$$

Since n is a strictly positive integer, this quantity can only be non-negative if $a \leq 0$. So that means any vector in $NC(u) \cap NC(v)$ must be in the set:

$$G_0' = \{g \in G_0 : g_i = g_j \leq 0\}$$

Let $w^\beta, \beta \neq i, j$ be the $(n-1)$-block partition vertex with $(w^\beta)_\beta = 1$ and $(w^\beta)_\alpha = -\frac{1}{n-1}$

for all $\alpha \neq \beta$, and choose some $g \in G_0'$ such that $g_\beta > 0$. Then evaluating the product of g with w^β versus u, we find:

$$\begin{aligned} g^T w^\beta &= g_\beta + a(-\frac{2}{n-1}) + \sum_{\alpha \neq i,j,\beta} g_\alpha \left(-\frac{1}{n-1}\right) \\ g^T u - g^T w^\beta &= \left(a(1 - \frac{1}{n-1} - \frac{1}{n-1} g_\beta\right) - \left(-\frac{2}{n-1} a + g_\beta\right) \\ &= (a - g_\beta)\left(1 + \frac{1}{n-1}\right) \end{aligned}$$

But since we have fixed $a \leq 0$ and $g_\beta > 0$, this quantity is guaranteed to be negative and therefore g cannot be in the normal cone of u (or by extension v). Since this is true for any g with $g_\beta > 0$ where $\beta \neq i, j$ is arbitrary, we must further restrict our set of vectors that can still be in the intersection to:

$$G_0'' = \{g \in G_0' : g_\alpha \leq 0 \quad \forall \alpha \in [d]\}$$

In other words, the two types of vertices we have looked at so far tell us that any vector $g \in \mathbb{R}^d$ that has even one positive coordinate $g_\alpha > 0$ cannot possibly be in the intersection of the normal cones of u and v. So we restrict our attention to vectors that are non-positive at all coordinates. Let us now consider the basal vertex $v_\omega = \left(-\frac{1}{n-1}, -\frac{1}{n-1}, ..., -\frac{1}{n-1}\right)$, representing the partition of n objects into n non-empty blocks of size 1 each. Then for any $g \in G_0''$, we can evaluate $g^T v_\omega$ and $g^T u - g^T v_\omega$:

$$\begin{aligned} g^T v_\omega &= a\left(-\frac{2}{n-1}\right) \sum_{\alpha \neq i,j} g_\alpha \left(-\frac{1}{n-1}\right) \\ g^T u - g^T v_\omega &= a\left(1 - \frac{1}{n-1}\right) - a\left(-\frac{2}{n-1}\right) \\ &= a\left(1 + \frac{1}{n-1}\right) \end{aligned}$$

But we have already had to restrict $a \leq 0$, to ensure $g^T u \geq g^T w^*$, which means the only way to get $g^T u \geq g^T v_\omega$ is to fix $a = 0$. However, doing so means we are now looking at some subspace of the vectors $g \in \mathbb{R}^d$ that have $g_i = g_j = 0$, which is now a space of dimension at most $d-2$. Therefore when u and v have disjoint size-2 blocks, the intersection of their respective normal cones has dimension at most $d-2$, meaning there is no edge between u and v.

On the other side, consider the case where u and v do not have disjoint size-2 blocks. In this scenario, i and j represent pairs of objects that are linked together, but there exists an object that is in both the pair represented by i and the pair represented by j. This means there is an associated object pair represented by some index h such that for any vertex w of $L_{n,n}$, $w_i = w_j = 1$ implies $w_h = 1$ as well. For this situation, we consider the following $(d-1)$-dimensional set of vectors and aim to prove that it is contained in $NC(u) \cap NC(v)$:

$$G_h = \{g \in \mathbb{R}^d : 0 \leq g_i = g_j < |g_h|, g_\alpha < 0 \quad \forall \alpha \neq i, j\}$$

The condition $g_\alpha < 0$ for all $\alpha \neq i, j$ means that $g_h < 0$ as well. Consider any $g \in G_h$. We will investigate several cases, to consider different types of vertices one step at a time.

Case 1: Consider the n-block partition vertex v_ω. We will have the same representation of $g^T v_\omega$ as in the previous part of the proof, which means $g^T u - g^T v_\omega$ looks the same as well:

$$g^T v_\omega = a(-\frac{2}{n-1}) \sum_{\alpha \neq i,j} g_\alpha \left(-\frac{1}{n-1}\right)$$

$$
\begin{aligned}
g^T u - g^T v_\omega &= a\left(1 - \frac{1}{n-1}\right) - a(-\frac{2}{n-1}) \\
&= a\left(1 + \frac{1}{n-1}\right)
\end{aligned}
$$

But this time, we are using $a > 0$, so we are guaranteed $g^T u \geq g^T v_\omega$.

Case 2: Let $w \neq u, v$ be an $(n-1)$-block vertex. Then there is a singular index $\beta \neq i, j$ such that $w_\beta = 1$, and we compute:

$$
\begin{aligned}
g^T w &= a(-\frac{2}{n-1}) + g_\beta + \sum_{\alpha \neq i,j,\beta} g_\alpha \left(-\frac{1}{n-1}\right) \\
g^T u - g^T w &= \left(a(1 - \frac{1}{n-1}) - \frac{1}{n-1} g_\beta\right) - \left(a(-\frac{2}{n-1}) + g_\beta\right) \\
&= a\left(1 + \frac{1}{n-1}\right) - g_\beta\left(1 + \frac{1}{n-1}\right)
\end{aligned}
$$

Since by definition $a > 0$ and $g_\beta < 0$, we get $g^T u \geq g^T w$.

Case 3: There are precisely two types of partitions with $n-2$ blocks, and in this proof we must consider them separately. For this section, let w represent an $(n-2)$-block partition with two size-2 blocks and all other blocks having size 1. Then w will contain exactly two 1s somewhere among its coordinates. Since u and v have overlapping size-2 blocks, the coordinate locations for these 1s can include **at most one** of i and j. We will separately look at the cases where these locations do or do not include i or j.

3a: First, suppose $w_i = 1$ (without loss of generality) and $w_\beta = 1$ for some $\beta \neq i, j$. Then evaluating $g^T w$ and $g^T u - g^T w$, we get:

$$
\begin{aligned}
g^T w &= a(1 - \frac{1}{n-1}) + g_\beta + \sum_{\alpha \neq i,j,\beta} g_\alpha \left(-\frac{1}{n-1}\right) \\
g^T u - g^T w &= -\frac{1}{n-1} g_\beta - g_\beta
\end{aligned}
$$

Since $g_\beta < 0$ for any $\beta \neq i, j$, this gives us $g^T u \geq g^T w$.

3b: Now, let the locations of the 1s in w be β, γ both distinct from i and j. Then when we compute $g^T w$ and $g^T u - g^T w$, we get:

$$\begin{aligned}
g^T w &= a(-\frac{2}{n-1}) + g_\beta + g_\gamma \sum_{\alpha \neq i,j,\beta,\gamma} g_\alpha \left(-\frac{1}{n-1}\right) \\
g^T u - g^T w &= \left(a(1 - \frac{1}{n-1}) - \frac{1}{n-1}(g_\beta + g_\gamma)\right) - \left(a(-\frac{2}{n-1}) + g_\beta + g_\gamma\right) \\
&= a(1 + \frac{1}{n-1}) - (g_\beta + g_\gamma)(1 + \frac{1}{n-1})
\end{aligned}$$

The prespecified conditions of $a > 0, g_\beta < 0, g_\gamma < 0$ guarantee that this expression is non-negative, so in this case we have $g^T u \geq g^T w$.

Case 4: Let w be a $(n-2)$-block partition vertex representing a partition with one size-3 block and all other blocks of size-1. One vertex in this category is $w = w^h$ representing the most refined mutual coarsening of u and v, which has 1s exactly at locations h, i, j. In the unique case of w^h, our computations give:

$$\begin{aligned}
g^T w^h &= 2a + g_h + \sum_{\alpha \neq h,i,j} g_\alpha \left(-\frac{1}{n-1}\right) \\
g^T u - g^T w^h &= \left(a(1 - \frac{1}{n-1}) - \frac{1}{n-1} g_h\right) - (2a + g_h) \\
&= -a(1 + \frac{1}{n-1}) - g_h(1 + \frac{1}{n-1})
\end{aligned}$$

For any $g \in G_h$, we have not only $a > 0, g_h < 0$ but also $g_i = a < |g_h|$. This final condition is what ensures $g^T u \geq g^T w^h$. Considering any other w from this case, we note that its three 1s must be at three locations that include **at most one** of i and j, because the three 1s together must represent a group of three objects that is **different** from the one given by h, i, j. But any such w will be dominated by at least one vertex from Case 3, and as such is dominated by u

and v since we have already proved Case 3.

Case 5: Finally, we consider the case where w is a vertex with $n-3$ blocks or fewer, meaning it is guaranteed to have at least three 1s among its indices. The indices containing these 1s may include neither, one, or both of i and j. However, since $g_\alpha < 0$ for all $\alpha \neq i, j$, there must exist w' in either Case 3 or Case 4 that dominates w. As such, it must also be true that $g^T u \geq g^T w$.

At the end of these cases, it has been shown that for any $g \in G_h$, $g^T u \geq g^T w$ for all vertices $w \neq u, v$ of $L_{n,n}$. Therefore G_h is contained in $NC(u) \cap NC(v)$, which means $NC(u) \cap NC(v)$ must have dimension at least $d-1$. Having much earlier shown that this intersection can have dimension at most $d-1$, we conclude that the intersection has dimension exactly $d-1$, and therefore an edge exists between u and v. □

The proofs of these propositions are greatly streamlined by the fact that regardless of n, there is only one type of $(n-1)$-block partition. For any other value of k, the k-block partitions will typically include multiple classes of partition; for example, in $n = 5$ and $k = 3$ we have both 3-1-1 partitions and 2-2-1 partitions, and as demonstrated in Section 3.3, the vertices corresponding to these different partition classes have different properties and behaviors. Nevertheless, the observations of Chapter 3 support the following more general conjecture:

Conjecture 4.1.11. *For any $n \geq 3, 2 \leq k < n$, let u, v be any two k-block partition vertices in $L_{n,k}$. Then an edge exists between u and v.*

This conjecture revises Proposition 4.1.9 for generalized k. Note that it is known to hold for $k = 2$ based on the previously-mentioned result of Lemma 2.3.5 in [6]; however, it is likely that neither the proof structure used in that paper nor the proof structure utilized for Proposition 4.1.9 will help much with other values of k, as they rely heavily on characteristics unique to the $k = 2$ polytope and to $(n-1)$-block partitions, respectively. Additionally, further inquiries are needed to characterize edges between partitions with m blocks when $k > m$. These would be generalizations of Proposition 4.1.10 for broader use.

4.2 Nested Iterations of $L_{n-m,k}$

In this section we prove and discuss our main result, which shows that some faces of $L_{n,k}$ are up-to-k-partition polytopes for $n-1, n-2, ..., n-m = k$. These faces can be defined according to the partitions represented by their incident vertices, indicating that the relationships between partitions exerts direct control over the low-dimensional structure of $L_{n,k}$. A different coordinate notation system will be utilized in this section, since these results rely strongly on the relationship between partition structure and the coordinate axes. Instead of representing a vector as $\mathbf{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d, d = \binom{n}{2}$, we will instead use a system based on the Gram matrix representation of partitions. Recall that a partition represented as a Gram matrix G is converted to a point in $\mathbb{R}^d$ by taking the unique elements $G_{ij}, j > i$, in order moving left-to-right across rows and from the top to the bottom of the matrix, resulting in a vector of the form:

$$(G_{12}, G_{13}, ..., G_{1n}, G_{23}, G_{24}, ..., G_{n-1,n}) \in \mathbb{R}^d$$

Correspondingly, in this section we will index any point in $\mathbb{R}^d$ as:

$$\mathbf{x} = (x_{12}, x_{13}, ..., x_{1n}, x_{23}, x_{24}, ..., x_{n-1,n}) \in \mathbb{R}^d$$

The next results suggest an eventual method of constructing $L_{n,k}$ from smaller polytopes. Much of the edge structure of $L_{n,k}$ that has not been previously elucidated can be understood through containment in these smaller polytopes. Together with the earlier results, this characterizes a significant portion of the edge structure for any $L_{n,k}$. Moreover, through these results we progress toward a broad understanding of the low-dimensional structure of $L_{n,k}$. We begin by showing that a particular set of faces related to partition structure must exist in $L_{n,k}$.

Lemma 4.2.1. *Let $2 \leq k < n$ and consider $L_{n,k}$, with vertices representing partitions of $[n]$ with up to k blocks. For any fixed pair of elements $a, b \in [n]$, let V_{ab} be the subset of vertices of $L_{n,k}$ representing partitions where a and b are in the same block. Then $L_{n,k}$ has a face supported on V_{ab}.*

Proof. Recall that a face of a polytope is defined to be any set of the form $P \cap \{\mathbf{x} \in \mathbb{R}^d : \mathbf{cx} = c_0\}$ where $\mathbf{cx} \leq c_0$ is a valid inequality for the polytope (Definition 2.1.2). By the Gram matrix structure of $L_{n,k}$, every vertex of $L_{n,k}$ satisfies $-1 \leq v_{ij} \leq 1$ regardless of n or k, and thus $x_{ab} \leq 1$ is a valid inequality for $L_{n,k}$. Therefore there exists some face F of $L_{n,k}$ such that $F = L_{n,k} \cap \{\mathbf{x} \in \mathbb{R}^d : x_{ab} = 1\}$. By the construction of $L_{n,k}$, for any vertex $v \in V_{ab}$, it must be true that $v_{ab} = 1$, and for any vertex $v \notin V_{ab}$, it is true that $v_{ab} < 1$. Thus V_{ab} is exactly the set of vertices incident on F. $\square$

Lemma 4.2.1 verifies and generalizes the observations made in Subsection 3.3.5 that 6-dimensional faces exist in $L_{5,k}$ with vertex sets that conjoin each pair of objects in [5]; this turns out to be true when considering partition polytopes for any n. The subset of partitions that define any such face within $L_{n,k}$ can be regarded as up-to-k partitions on a set of $n-1$ objects. For example, consider the $n = 5$ set of objects $\{a, b, c, d, e\}$. If we take the pair a and b, and restrict to partitions where these two objects are in the same block, we are essentially partitioning the $n = 4$ set of objects $\{ab, c, d, e\}$. In this view, a and b have been "glued together" to obtain a new, smaller set to partition.

To reach our main result regarding these faces, we first define an affine map between $\mathbb{R}^{\binom{n}{2}}$ and $\mathbb{R}^{\binom{n-1}{2}}$. For convenience of notation, set $e = \binom{n-1}{2}$.

Definition 4.2.2. For a fixed $a, b \in [n]$, the **partition transformation map** $\pi : \mathbb{R}^d \to \mathbb{R}^e, \mathbf{x} \mapsto A\mathbf{x}$ is defined by a d-by-e transformation matrix, with columns indexed like the axes of $\mathbb{R}^d$:

$$A = \begin{pmatrix} \vert & \vert & & \vert \\ \alpha_{12} & \alpha_{13} & \dots & \alpha_{n-1,n} \\ \vert & \vert & & \vert \end{pmatrix}$$

A column vector α_{ij} is the zero vector if i or j is equal to b; otherwise it is one of the standard basis vectors $\mathbf{e_m}, 1 \leq m \leq \binom{n-1}{2}$, for $\mathbb{R}^e$. Each basis vector is represented exactly once in the columns of A, and the basis vectors are ordered such that removing the zero-vector columns would yield the identity matrix.

The map π is designed to move from $\mathbb{R}^d$ to $\mathbb{R}^e$ by removing the specific $n - 1$ coordinate axes in $\mathbb{R}^d$ for which our fixed parameter b is one of the indices. We could equivalently remove those involving a, but only one of the two fixed parameters should be removed. The map is best understood with an example.

Example 4.2.3. Suppose $n = 4$, meaning $d = 6$ and $e = 3$, and fix $a = 1, b = 2$. Then the transformation matrix A is:

$$A = \begin{pmatrix} \vert & \vert & \vert & \vert & \vert & \vert \\ \alpha_{12} & \alpha_{13} & \alpha_{14} & \alpha_{23} & \alpha_{24} & \alpha_{34} \\ \vert & \vert & \vert & \vert & \vert & \vert \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The partition transformation map $\pi : \mathbb{R}^6 \to \mathbb{R}^3$ operates as follows on any column vector $\mathbf{x} \in \mathbb{R}^6$:

$$\pi(\mathbf{x}) = A\mathbf{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{12} \\ x_{13} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{34} \end{pmatrix} = \begin{pmatrix} x_{13} \\ x_{14} \\ x_{34} \end{pmatrix}$$

As desired, the elements of $\mathbf{x}$ retained by this operation are indexed solely by the labels 1, 3, and 4; the label 2 has been entirely removed, leaving only the coordinates that represent the relationships exclusively between these three of the original 4 objects in $[n]$.

Broadly, the map π removes the partition information for one object out of n, but retains all other partition information for the remaining $n-1$ objects. Since we are focused on circumstances where two objects a, b are positioned exclusively together with each other within partitions, this reduction in dimension can be seen as a removal of superfluous information.

We are now ready to prove the main theorem. Using π, we can show that the faces described in Lemma 4.2.1 are each affinely equivalent to the polytope $L_{n-1,k}$, which also ensures they are combinatorially equivalent to $L_{n-1,k}$.

Theorem 4.2.4. *Let $2 \leq k < n$ and consider $L_{n,k}$, with vertices representing partitions of $[n]$ with up to k blocks. For any fixed pair of elements $a, b \in [n]$, let V_{ab} be the subset of vertices of $L_{n,k}$ representing partitions where a and b are in the same block. Then the face F incident on V_{ab} is affinely, and thus combinatorially, equivalent to $L_{n-1,k}$.*

Proof. The partition understanding of $L_{n,k}$ heavily restricts the structure of F. Any vertex of $L_{n,k}$ incident on F has $v_{ab} = 1$, meaning a and b are in the same block. This means that for any $c \in [n], c \neq a, b$, the relationship between a and c must be identical to the relationship between b and c, and therefore $v_{ac} = v_{bc}$. Since this is true for all vertices of F, it is also true for any point $\mathbf{x} \in F$, meaning F is contained in an affine subspace of $\mathbb{R}^d$ characterized by $x_{ab} = 1, x_{ac} = x_{bc} \forall c \neq a, b$. This subspace has dimension $d - 1 - (n-2) = \frac{n(n-1)}{2} - n + 1 = \frac{(n-1)(n-2)}{2}$, so F considered as the convex hull of V_{ab} is a polytope of dimension at most $\frac{(n-1)(n-2)}{2} = \binom{n-1}{2}$. We denote the subspace as:

$$H_{ab} = \{x \in \mathbb{R}^d | x_{ab} = 1, x_{ac} = x_{bc} \text{ for any } c \neq a, b\}$$

Note that, being some affine subspace of $\mathbb{R}^d$ which contains F, H_{ab} must contain the affine hull of F. We now note that two polytopes P and Q in $\mathbb{R}^d$ and $\mathbb{R}^e$, respectively, are affine-equivalent if there exists an affine map $\pi : \mathbb{R}^d \to \mathbb{R}^e$ that maps P bijectively to Q, where π itself need not be a bijection but must restrict to a bijection between the affine hulls of the two polytopes [67]. We intend to prove that $F \subset \mathbb{R}^d$ is affine-equivalent to $L_{n-1,k} \subset \mathbb{R}^{\binom{n-1}{2}}$ using the partition transformation map π defined above.

Consider any vertex $v \in V_{ab}$, a vertex of the face $F \subset L_{n,k}$, and recall that v represents a valid partition of $[n]$ into at most k blocks by having $v_{ij} \in \{1, -\frac{1}{k-1}\}$ for any pair of objects $i, j \in [n]$. Then since the values v_{ij} are unaltered by π, and are purely either retained or discarded depending on whether b is one of the objects involved, it follows that $\pi(v)$ represents a partition of $[n] - \{b\}$ into up to k blocks. Therefore $\pi(v)$ must be one of the vertices of $L_{n-1,k} \subset \mathbb{R}^n$.

To progress further, it helps to establish that this transformation is bijective when restricted to H_{ab}, as this simplifies showing that it maps F to $L_{n-1,k}$. The most straightforward way to do this is to restrict the original mapping to $\pi_{ab} : H_{ab} \to \mathbb{R}^e$, where $\pi_{ab}(\mathbf{x}) = \pi(\mathbf{x})$ for any $\mathbf{x} \in H_{ab}$, and construct the inverse mapping $\pi_{ab}^{-1} : \mathbb{R}^e \to H_{ab}$. Note initially that H_{ab} also has dimension $\binom{n-1}{2} = e$. Our inverse will map $\mathbf{y} \mapsto C\mathbf{y} + \mathbf{d}$ for a carefully-chosen e-by-d matrix C and vector $\mathbf{d}$ of length d, indexed according to the axes of $\mathbb{R}^d$. The vector $\mathbf{d}$ will have $d_{ab} = 1$ and zeros everywhere else. The matrix is easiest to describe according to its row vectors, as these can be indexed according to the axes of $\mathbb{R}^d$ to align in structure with the original transformation matrix A:

$$C = \begin{pmatrix} — & c_{12} & — \\ — & c_{13} & — \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ — & c_{n-1,n} & — \end{pmatrix}$$

The row vector c_{ab} is the zero vector. For the row vectors c_{ij} such that neither i nor j is equal to b, each is equal to one of the standard basis vectors $\mathbf{e_m}$ of $\mathbb{R}^e$, such that if this set of vectors was taken in order as an e-by-e matrix, it would be the identity matrix. For the remaining row vectors, either i or j is equal to b, but not both; any such vector c_{ib}, c_{bj} is equal to the same standard basis vector as the corresponding row vector c_{ia}, c_{aj}.

We can now prove that π_{ab}^{-1} is in fact the inverse mapping of the restricted map π_{ab}. Consider any $\mathbf{x} \in H_{ab}$ and any $\mathbf{y} \in \mathbb{R}^e$; the maps are inverses if and only if $\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})) = \mathbf{x}$ and $\pi_{ab}(\pi_{ab}^{-1}(\mathbf{y})) = \mathbf{y}$. This can be accomplished on a coordinate-by-coordinate basis.

We first show that $\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})) = \mathbf{x}$. For any $\mathbf{x} \in H_{ab}$, it is true that $x_{ab} = 1$. Note that since c_{ab} is the zero vector and $d_{ab} = 1$ by definition, $(\pi_{ab}^{-1}(\mathbf{y}))_{ab} = 1$ for any $\mathbf{y} \in \mathbb{R}^e$. Since $\pi(\mathbf{x}) \in \mathbb{R}^e$ for any $\mathbf{x} \in \mathbb{R}^d$ and $\pi_{ab}(\mathbf{x}) = \pi(\mathbf{x})$ for all $\mathbf{x} \in H_{ab}$, it follows that $\pi_{ab}(\mathbf{x}) \in \mathbb{R}^e$ and thus $(\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})))_{ab} = 1$. We now turn to the coordinates ij where neither i nor j is equal to b. By construction, $(\pi_{ab}^{-1}(\mathbf{y}))_{ij} = \mathbf{e_m} \cdot \mathbf{y}$ for some standard basis vector $\mathbf{e_m}$, i.e. the m-th entry of $\mathbf{y}$ becomes $(\pi_{ab}^{-1}(\mathbf{y}))_{ij}$. However, for $\mathbf{y} = \pi_{ab}(\mathbf{x})$, exactly e entries of $\mathbf{x}$ are preserved in the order in which they appear in $\mathbf{x}$, and these are precisely those entries at the coordinates ij where neither i nor j is equal to b. Therefore for any $1 \leq m \leq e$, the m-th entry of $\pi_{ab}(\mathbf{x})$ is exactly the m-th entry of $\mathbf{x}$ to exist at a coordinate ij for which i and j are not equal to b, and so we conclude that for any such coordinate, $(\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})))_{ij} = x_{ij}$. Finally, consider the set of coordinates ij where exactly one of i and j is equal to b; without loss of generality, let it be $i = b$. Recall that for any $\mathbf{x} \in H_{ab}$ we have $x_{ac} = x_{bc}$ for all $c \neq a, b$. We have already established that for $j \neq b$, it must be true that $(\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})))_{aj} = x_{aj}$, as neither coordinate index is equal to b in this case. Consider the specific $\mathbf{e_m}$ such that $\mathbf{e_m} \cdot \mathbf{y} = (\pi_{ab}^{-1}(\mathbf{y}))_{aj}$; by construction, $(\pi_{ab}^{-1}(\mathbf{y}))_{bj} = \mathbf{e_m} \cdot \mathbf{y}$ for this same $\mathbf{e_m}$, meaning that $(\pi_{ab}^{-1}(\mathbf{y}))_{aj} = (\pi_{ab}^{-1}(\mathbf{y}))_{bj}$. Therefore $(\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x}))_{bj} = \pi_{ab}^{-1}(\pi_{ab}(\mathbf{x}))_{aj} = x_{aj} = x_{bj}$.

This covers all coordinates of $\mathbf{x}$, so we conclude that $\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})) = \mathbf{x}$.

Next we demonstrate that $\pi_{ab}(\pi_{ab}^{-1}(\mathbf{y})) = \mathbf{y}$ for any $\mathbf{y} \in \mathbb{R}^e$. Since $\mathbb{R}^e$ is the space which omits b, this is a much more straightforward proof. Note that for any $1 \le m \le e$, the m-th entry of $\mathbf{y}$ is copied by π_{ab}^{-1} into the m-th coordinate ij of $\pi_{ab}^{-1}(\mathbf{y})$ such that neither i nor j is equal to b. Moreover, for any $\mathbf{x} \in H_{ab}$, by construction the m-th entry of $\mathbf{x}$ among those whose coordinates ij satisfy the criterion that neither i nor j is equal to b is copied by π_{ab} into the m-th entry of $\pi_{ab}(\mathbf{x})$. Therefore we conclude that $(\pi_{ab}(\pi_{ab}^{-1}(\mathbf{y})))_m = \mathbf{y}_m$ for all $1 \le m \le e$, and thus $\pi_{ab}(\pi_{ab}^{-1}(\mathbf{y})) = \mathbf{y}$. Combining this with $\pi_{ab}^{-1}(\pi_{ab}(\mathbf{x})) = \mathbf{x}$, it follows that π_{ab} and π_{ab}^{-1} are inverses, and correspondingly that π_{ab} is a bijective mapping from H_{ab} to $\mathbb{R}^e$.

We demonstrated previously that any vertex of F is mapped to a vertex of $L_{n-1,k}$ by π_{ab}. Having shown that π_{ab} is a bijection, we can in fact conclude that any vertex of F is mapped to a unique vertex of $L_{n-1,k}$. The vertices of F represent the complete set of partitions of n items into at most k blocks such that the two specific items a, b are in the same block; therefore the number of vertices in V_{ab} is equal to the number of these partitions. Any partition of the set of $n-1$ items $[n]-\{b\}$ into up to k blocks, represented by a unique vertex of $L_{n-1,k}$, can be converted into a partition represented by a unique vertex in V_{ab} by adding the element b to the block of the partition containing a; therefore we conclude that the number of vertices in V_{ab} is exactly equal to the number of vertices of $L_{n-1,k}$, and thus the bijection π_{ab} maps the set of vertices of F to precisely the set of vertices of $L_{n-1,k}$. Since any point in F is an affine combination of its vertices, and likewise for $L_{n-1,k}$, it follows that $\pi_{ab}(F) = L_{n-1,k}$. Thus we have an affine map which takes F bijectively to $L_{n-1,k}$, and since H_{ab} contains the affine hull of F and likewise $\mathbb{R}^e$ contains the affine hull of $L_{n-1,k}$, our map must restrict to a bijection between the affine hulls of the polytopes. Therefore F and $L_{n-1,k}$ are affine-equivalent. In fact, $\mathbb{R}^e$ is exactly the affine hull of $L_{n-1,k}$ since $L_{n-1,k}$ is a polytope of dimension $\binom{n-1}{2} = e$, so therefore F is a polytope of dimension exactly $\binom{n-1}{2}$ as well.

Finally, we note that affine equivalence preserves the face lattice of a polytope. For any two faces $I, J \subseteq L_{n-1,k}$, $I \subseteq J$ if and only if $\pi_{ab}^{-1}(I) \subseteq \pi_{ab}^{-1}(J)$, and it is guaranteed that $\pi_{ab}^{-1}(I)$ and $\pi_{ab}^{-1}(J)$ are faces of F [74]. Since π_{ab} is an affine equivalence relation, it preserves the dimension of the faces, and therefore the containment property described above ensures that the face lattice of F is identical to the face lattice of $L_{n-1,k}$. As such, we can conclude that the face F defined by the vertex set V_{ab} is combinatorially equivalent to $L_{n-1,k}$. $\square$

To briefly clarify and expound on the inverse mapping $\pi_{ab}^{-1} : \mathbb{R}^e \to H_{ab}$ used in the proof, we can continue the explicit example used to demonstrate the partition transformation map. Using $n = 4$ and $a = 1, b = 2$, the inverse map starts with partition information for $n - 1 = 3$ and is intended to generate partition information that precisely matches the behavior of a, producing a new element b that exists in total alignment with a. In accordance with their definitions above, the transformation matrix C and vector d, along with the behavior of the map itself, are as follows over $\mathbf{y} \in \mathbb{R}^3$, with the elements of $\mathbf{y}$ labelled such that $b = 2$ is the omitted element:

$$C = \begin{pmatrix} — c_{12} — \\ — c_{13} — \\ — c_{14} — \\ — c_{23} — \\ — c_{24} — \\ — c_{34} — \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\pi_{ab}^{-1}(\mathbf{y}) = C\mathbf{y} + \mathbf{d} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_{13} \\ y_{14} \\ y_{34} \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ y_{13} \\ y_{14} \\ y_{13} \\ y_{14} \\ y_{34} \end{pmatrix}$$

Theorem 4.2.4 shows that not only do the faces described by Lemma 4.2.1 exist in $L_{n,k}$, but in fact they are all affinely equivalent to the polytope $L_{n-1,k}$. This means that the polytopal structure associated with partitions of $[n-1]$ is directly integrated into the structure for partitions of $[n]$, controlling a large portion of the lower-dimensional geometry of the polytope $L_{n,k}$. Since affine equivalence includes combinatorial equivalence, one consequence is that all face inclusion relationships are preserved, meaning that any $L_{n-1,k}$-equivalent face imposes relationships and structure on a subset of lower-dimensional faces of $L_{n,k}$ from $\binom{n-1}{2}$ dimensions down to one-dimensional edges. Moreover, we can take this notion of equivalence further and recursively find more faces at even lower dimensions that are similarly affine equivalent to smaller partition polytopes.

Corollary 4.2.5. *Suppose $2 \leq k \leq n-m$ for some integer $1 \leq m \leq n-2$ and consider $L_{n,k}$ with vertices representing partitions of $[n]$ with up to k blocks. For any fixed subset of $m+1$ elements $M \subset [n]$, let V_M be the subset of vertices of $L_{n,k}$ representing partitions where all elements in M are in the same block. Then $L_{n,k}$ has a face incident exactly on V_M, and this face is affinely, and thus combinatorially, equivalent to $L_{n-m,k}$.*

Proof. Note that Theorem 4.2.4 is the $m = 1$ case. Consider $m = 2$, and let

$M = \{a, b, c\}$ for some trio of distinct elements in $[n]$. Let F be the face combinatorially isomorphic to $L_{n-1,k}$ that is defined by $a, b \in M$, in the style of Theorem 4.2.4. Note that any partition vertex for which all elements in M are in the same block must be in F. Since F is combinatorially isomorphic to $L_{n-1,k}$ via the partition transformation map π, its vertices act as the partitions of the $(n-1)$-element set $[n] - \{b\}$. As such, the set of vertices of $L_{n,k}$ where all elements in M are in the same block is equal to the set of vertices of F where $\{a, c\}$ are in the same block; in other words, within F, M identifies the same partitions as $M - \{b\}$, a two-element subset of $[n] - \{b\}$. By Theorem 4.2.4, then, there exists a $\binom{n-2}{2}$-dimensional face F_M of F (and by extension of $L_{n,k}$) that is combinatorially equivalent to $L_{n-2,k}$ and whose vertices are exactly the vertices V_M.

Inductively assume the statement holds for $m = \ell - 1$, for some value ℓ such that $k \leq n - \ell$; this ensures $L_{n-\ell,k}$ is still a well-defined structure. Consider $m = \ell$, and some $M \subset [n]$ with $|M| = m$. For any subset of $M' \subset M$ of size $m - 1$, our inductive step indicates that there is a face F of $L_{n,k}$ which is incident exactly on the vertices representing partitions where all elements of M' are in the same partition block, and furthermore F is combinatorially equivalent to $L_{n-m+1,k}$. Then by the same logic as the $m = 2$ case, M acts as a size-2 subset of the elements whose partitions are represented by F, since all elements in M' are already glued together by the structure of F. Applying Theorem 4.2.4, we obtain a face $F_M \subset F$, which must also be a face of $L_{n,k}$, that is incident exactly on V_M and is combinatorially equivalent to $L_{n-m,k}$. □

Theorem 4.2.4 indicates that there will be several copies of $L_{n-1,k}$ present as faces of $L_{n,k}$; in fact, there will be one for each pair of elements in $[n]$, so there will be $\binom{n}{2}$ such faces overall. Similarly, the corollary indicates that there will be at least

$\binom{n}{m+1}$ copies of $L_{n-m,k}$ present as faces of $L_{n,k}$, defined by the subsets of $[n]$ of size $m+1$. Note that since every vertex of $L_{n,k}$ represents a partition of $[n]$ into at most k blocks, when $k < n$ it is ensured by the pigeonhole principle that at least one block of any partition will contain more than one element. Therefore, the corresponding vertices must belong to at least one copy of $L_{n-m,k}$, meaning every vertex of $L_{n,k}$ belongs to at least one such copy whenever $k < n$. This indicates that the structure of these copies has tremendous influence over the lower-dimensional structure of $L_{n,k}$, and especially over the edge structure. The lowest-dimension set of faces that can be utilized for this concept will be those equivalent to $L_{k,k}$.

Several open questions remain around nested faces. Corollary 4.2.5 considers them as defined by subsets of $[n]$; the eventual result of the progression described in the corollary is a collection of faces combinatorially equivalent to $L_{k,k}$ whose basal vertices are a specific subset of the k-block partitions of $[n]$. In particular, the $L_{k,k}$ faces described by Corollary 4.2.5 must have basal vertices that are $(n-k+1)-1-...-1$ partitions. In general, however, this is not the only possible k-block partition structure. For example, under $n = 5, k = 3$, Corollary 4.2.5 identifies a collection of $L_{3,3}$-equivalent faces whose basal vertices are 3-1-1 partitions, but makes no claims about 2-2-1 partitions. It is plausible, given observations and results thus far, that these other k-block partition classes would also give rise to $L_{k,k}$ subfaces defined instead by the relationships that "glue" certain objects to each other to result in a k-block partition.

Conjecture 4.2.6. *Every k-block partition vertex is the basal vertex of some face of $L_{n,k}$ that is combinatorially isomorphic to $L_{k,k}$.*

Similarly, it may be possible to use "gluing" relationships to define higher-dimensional faces as well, given enough room between n and k. The ability to identify and describe a broad range of faces would be useful in problems where partition polytopes arise as solution spaces; recall that partition polytopes are traditionally associated with cuts on a graph, a class of problems with close ties to optimization (many of which will be explored in the next section of the dissertation). Graph cut problems typically seek to cut a graph with many vertices into a few big pieces ($k << n$), meaning that the polytope will be very high-dimensional and much of its geometry may be influenced by these intermediate faces.

Finally, all results in this line of inquiry thus far keep k constant and look for faces that are combinatorially equivalent to polytopes using smaller values of n. However, it is possible that faces may also exist that are combinatorially equivalent to polytopes that use smaller values of k. This line of inquiry should be fully explored, beginning by returning to $n = 4$ and $n = 5$ case studies to look for any evidence of such faces.

Part II

On A Max-Cut Algorithm

Chapter Five

Background: Max-Cut Problems and Semidefinite Programming

For this part of the dissertation, we assume a foundational understanding of optimization in general and combinatorial optimization in particular, per the books and notes of Bertsimas [10] and Schrijver [59]. The *k-max-cut problem* is a classic combinatorial optimization problem on weighted graphs. For any graph $G = (V, E)$ on n vertices, a k-cut is a partition of the vertices into k groups, and the weight of that cut is the total weight of all edges passing between groups. The solution to the k-max-cut problem on a graph is the k-cut with the largest weight. Letting $\mathcal{P} = B_1, ..., B_\ell$ be a ℓ-block partition of the graph vertices, some of whose blocks may be empty, we denote the number of non-empty blocks of the partition as $|\mathcal{P}|$. Then the problem can be formulated as follows, where w_{ij} is the weight of the edge between vertices i and j: [31]:

$$\begin{aligned} &\text{Maximize: } \sum_{1 \leq r < s \leq \ell} \sum_{i \in P_r, j \in P_s} w_{ij} \\ &\text{Subject to: } |\mathcal{P}| = k \end{aligned}$$

These problems are NP-hard, so algorithms prioritize efficiently finding a reasonable approximation rather than an exact solution [32]. Goemans and Williamson famously developed a relaxation of the max-cut problem for $k = 2$ that guarantees a cut weight of at least 0.87856 of the true optimum for any graph [32]. There is substantial evidence that this is the best approximation guarantee any algorithm can achieve for the $k = 2$ max-cut problem [38, 42]. However, attempts to expand their methods for higher values of k found that the guarantee rapidly deteriorates as k increases [31].

5.1 The Goemans-Williamson approach

The Goemans-Williamson algorithm relaxes the 2-max-cut problem on a graph with n vertices to a *semidefinite program* (SDP): a linear objective function to be maximized, accompanied by a set of matrix constraints, whose solution will be a n-by-n positive semidefinite (PSD) matrix. The initial formulation that leads to the relaxation is an integer quadratic problem over variables $y_i, 1 \leq i \leq n$, where w_{ij} is the weight of the edge between vertex i and vertex j:

$$\text{Maximize: } \frac{1}{2}\sum_{i<j} w_{ij}(1 - y_i y_j)$$

$$\text{Subject to: } y_i \in \{-1, 1\} \quad \forall i \in V$$

The semidefinite relaxation of this problem replaces the integer variables with vectors v_i in the n-dimensional unit sphere S_{n-1} and considers their inner products [32]:

$$\text{Maximize: } \frac{1}{2}\sum_{i<j} w_{ij}(1 - v_i \cdot v_j)$$

$$\text{Subject to: } v_i \in S_{n-1} \quad \forall i \in V$$

As a result of this change in structure, the solution to the 2-max-cut SDP will be a *Gram matrix* (see Definition 2.3.2) whose entries A_{ij} are each the dot product of a pair of unit vectors i, j from a collection of n unit vectors [46]. Recall that by construction, any Gram matrix will belong to the space of n-by-n symmetric matrices $\mathcal{S}(n)$, and will be positive semidefinite. In this specific problem, since the Gram matrices are constructed using a collection of unit vectors, the diagonal entries of any feasible solution matrix will all be equal to 1. Thus the solution space for this problem is the $\binom{n}{2}$-dimensional convex body made up of these special Gram matrices,

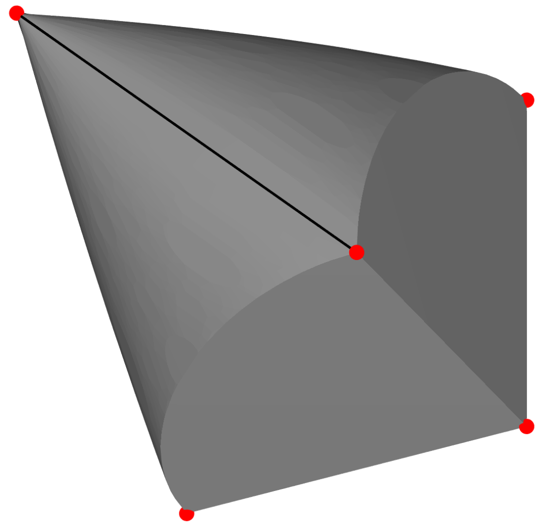


Figure 5.1: The elliptope $\mathcal{L}_{3,3}$, whose extremal points (red) represent partitions of 3 objects into at most 3 non-empty groups. Some regions of the elliptope's surface are affine and resemble polytope faces, while other regions are puffy or curved.

which is known as the elliptope $\mathcal{L}_n$:

$$\mathcal{L}_n := \{X \in \mathcal{S}(n) : X \succeq 0, X_{ii} = 1\}$$

There are 2^n extreme points of $\mathcal{L}_n$; each is a matrix whose entries are all either 1 or -1. Each extreme point can be interpreted as a division of n objects into 2 groups [47]. In other words, the extreme points of $\mathcal{L}_n$ are the discrete solutions to the 2-max-cut problem. For higher values of k, the SDP has a solution space that retains this key property by intersecting $\mathcal{L}_n$ with an orthant, yielding the k-way elliptope $\mathcal{L}_{n,k}$:

$$\mathcal{L}_{n,k} := \left\{ X \in \mathcal{L}_n : X_{ij} \geq -\frac{1}{k-1} \right\}$$

For $k = 2$, $\mathcal{L}_{n,k} = \mathcal{L}_{n,2} = \mathcal{L}_n$. $\mathcal{L}_{n,k}$ exhibits both polytopal and non-polytopal boundary characteristics, as can be seen in Figure 5.1. The k-way elliptope is a convex space whose extreme points represent the partitions of n objects into at most k non-empty groups [29]. These are the same partitions as discussed in Section 2.2, and in fact the extreme points of $\mathcal{L}_{n,k}$ are exactly the vertices of the up-to-k-partition polytopes discussed in the first part of the dissertation. The number of extreme points of $\mathcal{L}_{n,k}$ is equal to the total number of up-to-k-partitions of n as discussed in Section 2.2 and can be represented as a sum of Stirling numbers of the second kind:

$$\# \text{ of extreme points of } \mathcal{L}_{n,k} = \sum_{i=1}^{k} S(n,i)$$

The Goemans-Williamson algorithm includes a randomized rounding scheme for obtaining a discrete solution to the 2-max-cut problem from the SDP result. In $k = 2$ this scheme works extremely well due to the many nice symmetries of $\mathcal{L}_n$. For $k > 2$, however, $\mathcal{L}_{n,k}$ lacks many of these nice symmetries. As a result, the approximation guarantees gradually worsen as k increases, becoming arbitrarily poor [31]. This motivates the use of alternative rounding schemes for $k > 2$ cases.

5.2 Rounding with Iterated Linear Optimization

In 2021, Felzenszwalb, Klivans, and Paul proposed a deterministic algorithm for rounding the result of an SDP to an extremal point, dubbed *iterated linear optimization* (ILO) [29]. Broadly, this process can be exercised over any compact domain $\Delta \subset \mathbb{R}^n$. From any initial starting point $x_0 \in \Delta$, such as the solution obtained by solving an SDP, the algorithm constructs a sequence $x_1, x_2, ... \in \Delta$ by maximizing over an inner product:

$$x_{i+1} = T(x_i) = \operatorname*{argmax}_{y \in \Delta} x_i \cdot y$$

Thi sequence continues until a *fixed point* x^* is encountered.

Definition 5.2.1. For any transformation $T(x)$, a **fixed point** x^* is a point at which

$x^* = T(x^*)$.

In general, fixed points can be identified according to normal cones. Recall from Definition 2.1.8 that the normal cone of a point x in a convex body Δ is given by:

$$NC_\Delta(x) := \{y \in \mathbb{R}^n : x \cdot y \geq z \cdot y \quad \forall z \in \Delta\}$$

For $T(x)$, a point x^* is a fixed point if and only if x^* resides in its own normal cone, i.e. $x^* \cdot x^* \geq z \cdot x^*$ for all $z \in \Delta$ [29]. The process is always guaranteed to converge to a fixed point over any convex body Δ [29].

Importantly for the k-max-cut setting, when using $T(x)$ over a k-way elliptope $\mathcal{L}_{n,k}$, all extremal points of $\mathcal{L}_{n,k}$ are fixed points of this iterative construction [29]. However, the extremal points of the elliptope are not the only fixed points of T, as shown in Figure 5.2. The extremal points are rank 1 matrices, and they are the only rank 1 matrices in $\mathcal{L}_n$, which means they are the only points to have full-dimensional normal cones like polytope vertices; but even under this condition, $T(x)$ has other fixed points that may influence its behavior. This motivates a broader investigation of where and how ILO converges on various convex bodies.

There are two primary categories of fixed point behavior: attractive and repulsive. For the purposes of this inquiry, we will use similar definitions of attractive and repulsive fixed points as [29].

Definition 5.2.2. A fixed point x^* is **attractive** if there exists some $\epsilon > 0$ such that $||x^* - x_0|| < \epsilon$ implies that iteration with T starting at x_0 converges to x^*.

Definition 5.2.3. A fixed point x^* is **repelling** if for any $\epsilon > 0$, the condition $||x^* - x_0|| < \epsilon$ where $x_0 \neq x^*$ implies there is an n for which iteration with T starting at x_0 leads to x_n with $||x_n - x^*|| > \epsilon$.

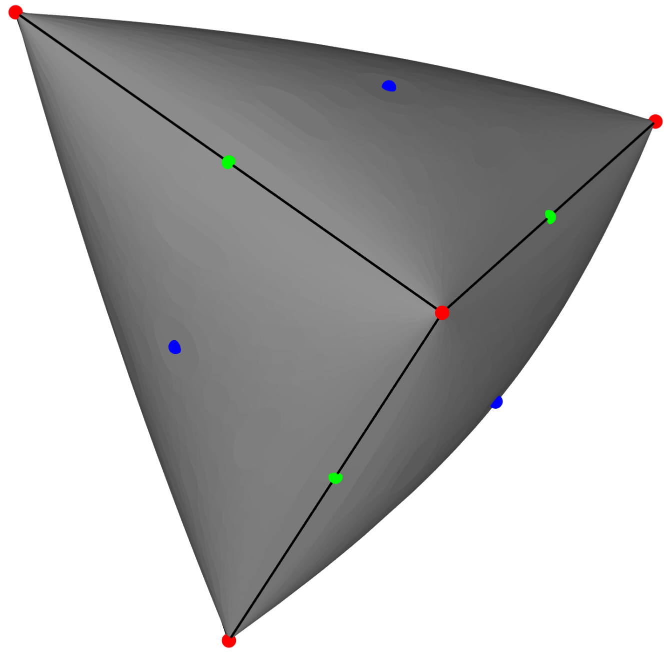


Figure 5.2: The fixed points of iterated linear optimization on the $n = 3$ elliptope $\mathcal{L}_3$, color-coded by type. Red fixed points are irreducible matrices of rank 1, which have 3-dimensional normal cones. Green fixed points are reducible matrices of rank 2, and blue fixed points are irreducible matrices of rank 2. [29]

For an attractive fixed point, as long as $T(x)$ starts close enough to the fixed point, it will converge to the fixed point. For a repelling fixed point, no matter how close to that fixed point $T(x)$ starts (as long as it does not start exactly on the fixed point), repeated iterations will move away from the fixed point and converge elsewhere. It is apparent from the work of [29] that the location of a convex body in space influences fixed point locations and behavior. Further investigation of this interaction will support an understanding of how ILO interacts not only with the elliptope for semidefinite programming applications, but with solution spaces to other relaxations of combinatorial problems as well. Chapter 6 investigates ellipses and polytopes as settings for $T(x)$, and assembles some results on the types and locations of the fixed points of $T(x)$ in these contexts.

5.3 Applications to Clustering

In data science, a *clustering problem* on a set of n data points seeks a partition of the data into an unknown number of groups that should reflect some natural sorting of the original dataset. Broadly, "natural" means that data points which exhibit similar characteristics ought to be put in the same cluster; there are many ways to codify this objective mathematically. However, clustering is typically used for large volumes of high-dimensional data, meaning n is very large and the data cannot be easily visualized to estimate the number of natural groups. Combining the high volume and dimension of the data with the number of ways to partition n, previously established to be a sum of Stirling numbers of the second kind, it becomes clear that the solution space for any clustering problem is extremely large. Many algorithmic approaches have been developed to address this complexity, using a variety of theoretical frameworks; for more details on common clustering algorithms and their theoretical underpinnings, we recommend *Algorithm Design* by Kleinberg and Tardos [44].

When clustering data, the dataset is often equipped with some quantitative representation of *similarity* between data points. Often, similarity is expressed as a non-negative value giving some notion of directionless distance between data points. The value is zero if and only if two points are identical, with greater distances indicating less similarity between points. By quantifying similarity, any clustering problem can be reimagined as a max-cut problem: the data can be considered as the vertices of a complete graph, with edges weighted by the measure's similarity values for each pair of data points. Considering the clustering problem in this way allows us to apply techniques used for the max-cut problem to clustering as well.

In [30], Felzenszwalb, Klivans, and Paul explore the prospect of implementing ILO for clustering, comparing it to the Goemans-Williamson algorithm for $k > 2$. The study demonstrates the potential of ILO not only as an alternative rounding scheme for semidefinite programming in the k-max-cut setting, but also as an algorithm for data clustering. To delve deeper into the possibility of using semidefinite programming with ILO for clustering, Chapter 7 explores a variety of simulations carried out to compare the performance of this algorithm to k-means, a commonly-used clustering algorithm in industry settings. The data for the experiments represents a wide array of conditions around the true number of clusters in a dataset, the spread and messiness of the data, and the dimension of the data. The comparison utilizes multiple scoring approaches to assess the performances of both algorithms under various circumstances, the details of which are presented in Subsection 7.1.2. Based on the outcomes of the clustering experiments, a few specific areas of study are recommended for future work regarding semidefinite programming and ILO as an approach to clustering.

Chapter Six

Fixed Points and Behavior

We undertook assortment of simulations and computations to explore the behavior of $T(x)$ on various convex bodies situation in $\mathbb{R}^n$, beginning with models in 2 dimensions. The main objective of this investigation was to expand on the results of [29] with regards to the fixed points of $T(x)$ on various convex domains. Since the ultimate domains of interest are the elliptopes $\mathcal{L}_{n,k}$ discussed in Section 5.1, this was divided into two main areas of inquiry. Elliptopes have boundaries with both polytopal and non-polytopal characteristics; since these two types of boundaries yield dramatically different algorithm behaviors in, the inquiry is split into two sections focused on both boundary types individually.

First, in Section 6.1, fixed points are characterized on two-dimensional ellipses, focusing on how the specific location of a convex body in $\mathbb{R}^n$ affects the locations of the fixed points. For non-polytopal bodies, this influence is a continuous relationship. Understanding this behavior is important because while $\mathcal{L}_{n,2}$ has several symmetries around the origin, $\mathcal{L}_{n,k}$ generally has far fewer, which may impact the algorithm's behavior in the k-way case.

Second, in Section 6.2, fixed points are investigated on polytopes in two and three dimensions. We obtain partial results to generalize the observations for higher dimensions. The goal is the identification of behavioral characteristics that predict how the algorithm will navigate the partition polytopes discussed in Chapter 3 and Chapter 4. The section focuses in particular on the identification of *saddle points* among the polytope fixed points, as a proof-of-concept that such fixed points for the algorithm could exist.

6.1 Ellipses

An *ellipse* in two dimensions is a convex body defined by its center point $(h, k) \in \mathbb{R}^2$ and a pair of axis lengths $a, b > 0$ representing the maximal and minimal distances, respectively, of any point on the ellipse from the center. Using these parameters, a horizontal ellipse is given by the equation:

$$\frac{(x_1 - h)^2}{a^2} + \frac{(x_2 - k)^2}{b^2} = 1$$

The *major axis* a of this body is parallel to the x-axis and the *minor axis* b is parallel to the y-axis. Ellipses are conic forms, obtained by intersecting a three-dimensional cone with a plane.

To identify fixed points on the ellipse, we must identify any points $\mathbf{x} = (x_1, x_2)$ on the ellipse such that $T(\mathbf{x}) = \mathbf{x}$. Since $T(\mathbf{x})$ searches for the point in a convex body that maximizes the inner product, in this case the Euclidean inner product, this must be a point $\mathbf{x}$ such that over all $\mathbf{y} = (y_1, y_2)$ on the ellipse:

$$x_1^2 + x_2^2 \geq x_1 y_1 + x_2 y_2$$

The ellipse is highly regular, and as such a convenient approach to identifying these points is to parametrize the surface so that it is described in terms of a single variable t rather than two variables x_1, x_2. Any point on the ellipse can alternatively be represented as $(a \cos t + h, b \sin t + h)$ for some angle $0 \leq t < 2\pi$, which means the iterative algorithm has become:

$$T(t) = \underset{0 \leq s < 2\pi}{\operatorname{argmax}}\, a^2 \cos t \cos s + ah(\cos t + \cos s) + h^2 + b^2 \sin t \sin s + bk(\sin t + \sin s) + k^2$$

The simplest way to find the value of s that maximizes this expression is to take the derivative with respect to s and evaluate for where the derivative is 0. Taking the derivative removes several terms from the expression, at which point we solve to separate $s = T(t)$ from t:

$$\begin{aligned} 0 &= -a^2 \cos t \sin s - ah \sin s + b^2 \sin t \cos s + bk \cos s \\ &= -\sin s(a^2 \cos t + ah) + \cos s(b^2 \sin t + bk) \\ \sin s &= \frac{b^2 \sin t + bk}{a^2 \cos t + ah} \cos s \\ \tan s &= \frac{b^2 \sin t + bk}{a^2 \cos t + ah} \end{aligned}$$

Fixed points must be the points at which $t = T(t)$, so we conclude that the necessary condition for a point on the ellipse to be a fixed point is:

$$\tan t = \frac{b^2 \sin t + bk}{a^2 \cos t + ah}$$

Note that this is only the condition for the expression inside the maximization to be 0. Some values of t which satisfy this condition will maximize the expression, becoming attractive fixed points; some will instead minimize it and act as repulsive fixed points. The existence of these attractive and repulsive fixed points align with the observations made in [29] on the fixed points when iterating over a circle.

Considering ellipses in $\mathbb{R}^2$ more broadly, it can also be noted that an ellipse may not always have its major and minor axes aligned with the horizontal and vertical axes. Suppose an ellipse has been rotated from a horizontal position by some angle $0 \leq \theta < \pi$ (a rotation of $\theta = \pi$ will simply switch which ends of the horizontal axis point left and right). Independent from the location of the ellipse in the plane, in essence supposing a center of $(h,k)=(0,0)$, a given point (x_1,x_2) on the ellipse is relocated under such a rotation to a new point (x_1',x_2') by the following transformation:

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1\cos\theta - x_2\sin\theta \\ x_1\sin\theta + x_2\cos\theta \end{pmatrix}$$

Applying the earlier change of variables and adjusting for a generalized center, we have the following form for the ellipse, where a,b,h,k,θ are fixed-value parameters:

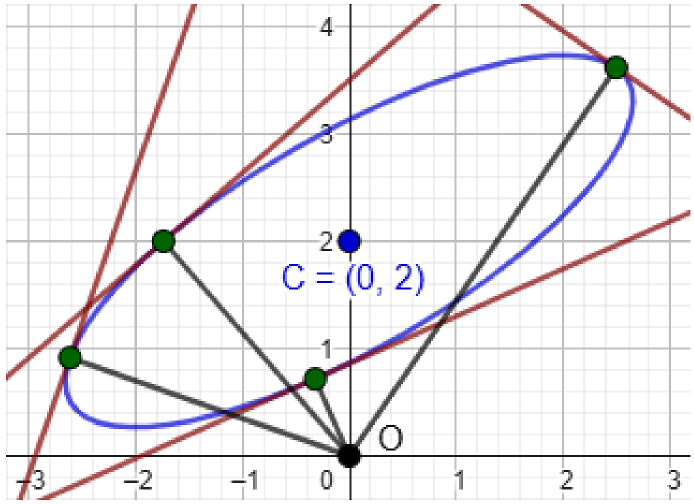
$$(a\cos t\cos\theta - b\sin t\sin\theta + h, a\cos t\sin\theta + b\sin t\cos\theta + k) \in \mathbb{R}^2, \quad 0 \le t < 2\pi$$

And from here, we can utilize the same process as for the horizontal ellipse to determine the criterion for fixed points, which results in:

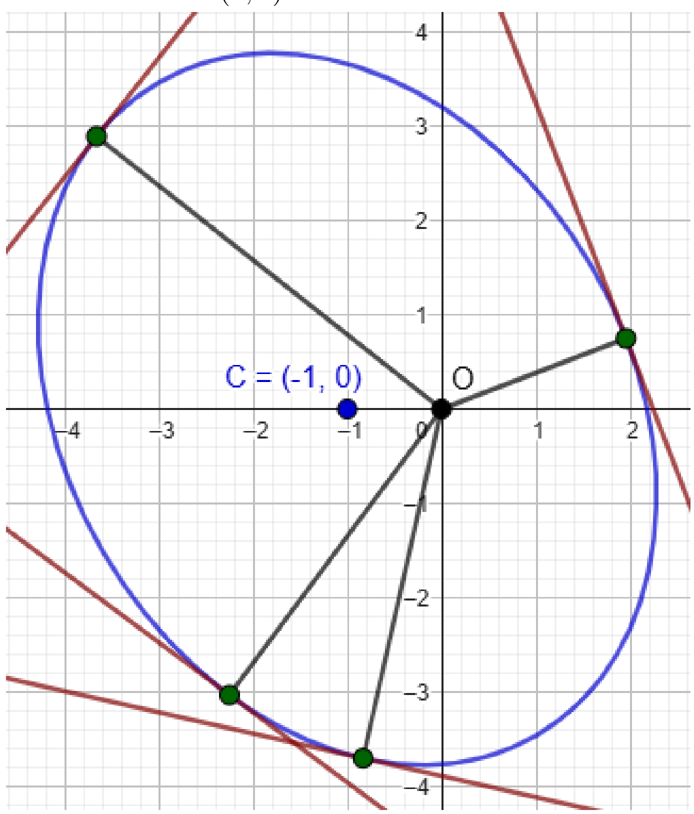
$$\tan t = \frac{b^2\sin t + b(k\cos\theta - h\sin\theta)}{a^2\cos t + a(h\cos\theta + k\sin\theta)}$$

As a quick but effective sanity-check of this condition, it can be observed that a value of $\theta = 0$ returns the prior result.

From these equations, it is evident that the locations of fixed points on an ellipse are governed not solely by the geometry of the ellipse, but by the location of the ellipse in $\mathbb{R}^2$. This is consistent with the results observed in [29] for circles, indicating that the position of a convex body with respect to the origin is likely to consistently influence the behavior of this algorithm. This is reinforced by graphical analysis, plotting the locations of points on the ellipse satisfying the identified conditions for an assortment of centers, axes, and rotation angles.



(a) An ellipse with parameters $a = 3, b = 1, \theta = \frac{\pi}{6}$ centered at $C = (0, 2)$.



(b) An ellipse with parameters $a = 4, b = 3, \theta = \frac{2\pi}{3}$ centered at $C = (-1, 0)$.

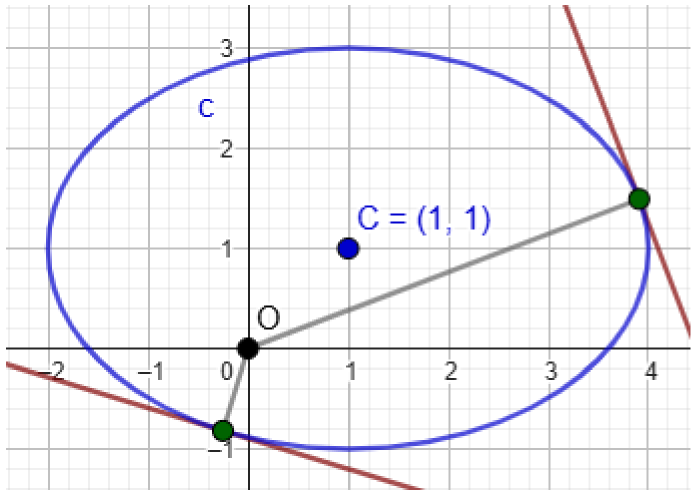
Figure 6.1: Two ellipses, drawn in blue and with their centers C marked in blue. Each has 4 fixed points (green). The origin O is marked in black, with gray line segments to each fixed point. The tangent lines to the ellipse at each fixed point are shown in red and are perpendicular to the gray segments.

Although both of the ellipses shown above exhibit 4 fixed points, the number of fixed points that appear on the ellipse can vary. Some ellipses, most commonly those near the origin, have 4 distinct fixed points; others, most commonly those farther from the origin, have only 2. However, the trigger for this behavioral shift proved to be elusive. Containment of the origin – or even proximity to the origin – are not correlated, for example. Many ellipses near the origin have 4 fixed points, but there are clear examples of ellipses containing the origin and having only 2 fixed points (such as Figure 6.2a) and of ellipses having 4 fixed points despite not containing the origin (Figure 6.1a, Figure 6.2b). Similarly, there is no correlation of numbers of fixed points with the distance of the ellipse's center from the origin, the ratio between the major and minor axes, or whether the ellipse intersected one of the axes of $\mathbb{R}^2$. The question of the overarching paradigm for the number of fixed points on the ellipse remains open.

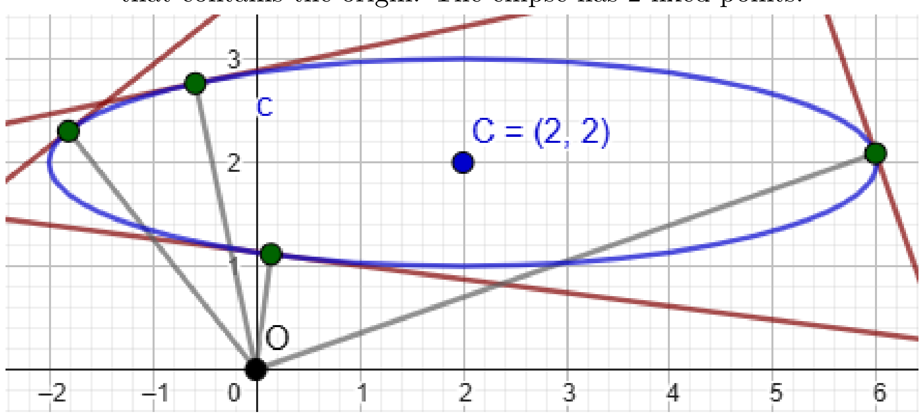
6.2 Polytopes

We begin the inquiry into polytopes by establishing some intuition for the behavior of $T(x)$ along one-dimensional affine structures, in particular edges of polytopes. Let $\operatorname{sgn}(x)$ denote the "sign" operation, which returns whether the input value is above, below, or equal to 0.

Lemma 6.2.1. *Suppose an n-dimensional polytope P has pair of vertices v, w connected by an edge such that only one coordinate value changes along that edge, i.e. there exists exactly one value $i \in [n]$ such that $|v_i - w_i| \neq 0$. Then $T(x)$ applied to*



(a) An ellipse centered at $C = (1, 1)$ with $a = 3, b = 2$ that contains the origin. The ellipse has 2 fixed points.



(b) An ellipse centered at $C = (2, 2)$ with $a = 4, b = 1$ that does not contain the origin. The ellipse has 4 fixed points.

Figure 6.2: The number of fixed points on a two-dimensional ellipse is not correlated with containment of the origin. The center of each ellipse is marked in blue, and the fixed points are marked in green. The gray line segments in each figure mark the lines from the origin to each fixed point, and the red lines are the tangent lines to the ellipse at each fixed point. Each tangent line is perpendicular to a gray segment.

any point x on that edge, with the edge itself as the domain, will operate as follows:

$$T(x) = \begin{cases} \underset{y \in \{v,w\}}{\operatorname{argmax}} \, ||y||, & \operatorname{sgn}(x_i) = \operatorname{sgn}(v_i) = \operatorname{sgn}(w_i) \\ v, & \operatorname{sgn}(x_i) = \operatorname{sgn}(v_i) \neq \operatorname{sgn}(w_i) \\ w, & \operatorname{sgn}(x_i) = \operatorname{sgn}(w_i) \neq \operatorname{sgn}(v_i) \end{cases}$$

Proof. The iterative process $T(x)$ operates by maximizing the inner product of x with all possible points y in the domain. By restricting the domain to the edge itself, the end result of $T(x)$ must be one of the two endpoints, v or w. The inner product of x with any point y will be the sum of the products of their elements, and along the edge between v and w, these products will all be the same except for $x_i y_i$: the product at the index which is changing along the edge. Therefore, the output of $T(x)$ will be the point y such that $x_i y_i$ is maximized. Note that regardless of anything else, either $v_i > y_i > w_i$ or $w_i > y_i > v_i$ for all points y along the edge. So either v or w must be the maximizer.

There are several cases to consider here. First, suppose that v_i and w_i have the same sign. If x_i matches them, then $x_i y_i$ is greatest at whichever endpoint has the greater value at index i, which is an equivalent condition to identifying which endpoint has the greater norm. If x_i has a different sign, however, then $x_i y_i < 0$ for every point y along the line, and so this term will detract from the total inner product. Thus the inner product will be maximized at whichever endpoint has the smaller value at index i, i.e. which has the smaller norm. On the other hand, if v_i and w_i have different signs, then x_i must match either v_i or w_i in sign. Consider the case where it matches v_i; then $x_i v_i > 0$ and $x_i w_i < 0$, meaning $x_i v_i$ adds to the inner product while $x_i w_i$ subtracts from it, and therefore the inner product will be maximized at v.

If x_i instead were to match w_i, these relationships would be exchanged, giving the result. □

Lemma 6.2.1 is applicable to a specific circumstance in the partition polytopes described in Chapter 3 and Chapter 4. We proved in Proposition 4.1.3 that in these polytopes, an edge exists between any two partition vertices related by the division of a single size-2 block into a pair of size-1 blocks. The coordinate-wise structure of the partition vertices means that this pair of vertices appear identical except at a single coordinate index, and furthermore that this index will be positive for the vertex with the size-2 block and negative for the other vertex. By this lemma, we fully understand the behavior of $T(x)$ along the guaranteed edge between any two such vertices. Moreover, the restriction of the domain of $T(x)$ to the edge is sensible in this context, because the algorithm would typically start from the interior of the polytope and step to the boundary. In other words, the edge point would be the result of some iteration of T, rather than an arbitrary starting point.

Lemma 6.2.1 demonstrates an overarching theme for these polytope inquiries: determining not just where fixed points are, but how they behave. When there are multiple attractive fixed points in a single convex body Δ, different starting points x_0 for the iterative algorithm $T(x)$ can converge to different fixed points. Knowing which fixed points a given starting point will move towards or away from is crucial to a broad understanding of the algorithm. To that end, we introduce the following terminology:

Definition 6.2.2. The **region of optimality** for a fixed point x^* of $T(x)$ over some convex body Δ is the set $C_{x^*} \subset \mathbb{R}^n$ such that starting the algorithm at any point $x \in C_{x^*} \cap \Delta$ can converge to x^*.

The definition uses "can converge" rather than "will converge" to account for the fact that $T(x)$ does not necessarily have a unique solution [29]. Note that to be a fixed point in the first place, x^* must reside within its own normal cone (Definition 2.1.8), which is equivalent to satisfying $x^* = T(x^*)$. Therefore the region of optimality for any fixed point will always contain x^* itself. However, in the case of a repulsive fixed point, the region of optimality will consist solely of x^* itself. For an attractive fixed point, on the other hand, the region of optimality must contain the entire normal cone of x^*. By the definition of the normal cone, any point in the normal cone has its inner product over Δ maximized by x^*. The intersection of Δ with the normal cone of x^* thus consists exactly of the points in Δ that will be drawn to x^* in a single step of $T(x)$. In the case of a polytope P, for which $\mathbb{R}^d$ is divided among the normal cones of the vertices, any point $x \in P$ belongs to the normal cone of at least one vertex v and will therefore have $T(x) = v$; hence, whenever a vertex v of a polytope resides in its own normal cone, it is a fixed point whose region of optimality is exactly its normal cone.

Moving forward, we can introduce the central question of the section. In addition to attractive and repulsive fixed points, there is a third category of fixed points called *saddle points*, which are partially attractive and partially repulsive. Let $\mathcal{B}_\varepsilon(x) \subset \mathbb{R}^d$ be a ball of radius ε centered at $x \in \mathbb{R}^d$, for some arbitrarily small $\varepsilon > 0$. The following definition can be utilized for fixed points of any optimization algorithm.

Definition 6.2.3. A fixed point x^* is called a **saddle point** if there is some $\varepsilon > 0$ such that there exists a set of points $A \subset \mathcal{B}_\varepsilon(x^*)$ and a set of points $R \subset \mathcal{B}_\varepsilon(x^*)$ for which:

(1) For any $x_0 \in A$, $||x - x_0|| < \varepsilon$, iteration with T starting at x_0 will converge to

x^*; and,

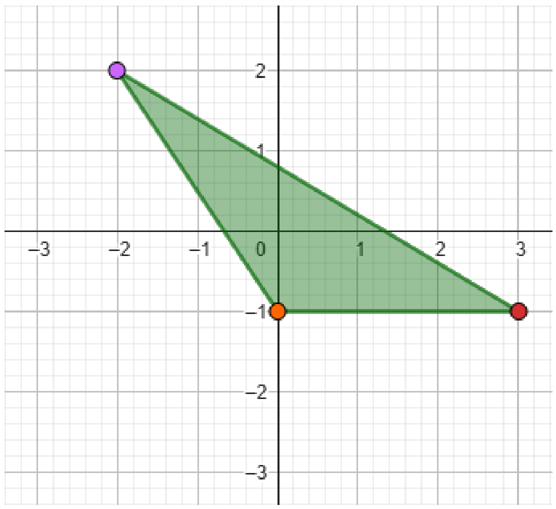
(2) For any $x_0 \in R$, $||x - x_0|| < \varepsilon$, iteration with T starting at x_0 will lead to some point x_n with $||x - x_n|| > \varepsilon$.

It was established in [29] that some convex bodies have fixed points for $T(x)$ which are saddle points, but for which either the region of attraction A or the region of repulsion R is not full-dimensional, i.e. n-dimensional. For example, a saddle point might repel $T(x)$ along a specific line on the surface of the convex body, but attract $T(x)$ from anywhere else on the surface. An open question stemming from this is whether it is possible for $T(x)$ to have saddle points whose regions of attraction and repulsion are both full-dimensional. We can characterize these points as follows:

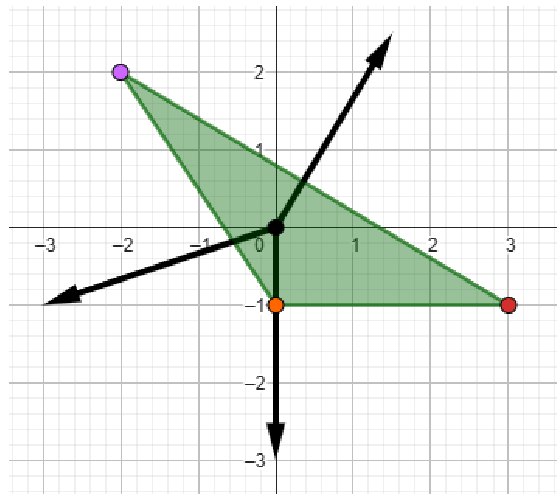
Definition 6.2.4. A saddle point s^* of $T(x)$ over some convex body $\Delta \subset \mathbb{R}^d$ is called **full-measure** if for any small neighborhood $N_\varepsilon = \mathcal{B}_\varepsilon(x) \cap \Delta$ of s^* within Δ, where $A, R \subset N_\varepsilon$ are the subsets of points in the neighborhood which are respectively attracted to and repelled from x, A and R are both d-dimensional.

Note that to be a full-measure saddle point, s^* must be a fixed point of $T(x)$ (i.e. reside in its own normal cone, as established in [29]) and have a d-dimensional region of optimality to which A belongs. We know that the region of optimality for any fixed point x^* must contain its normal cone. As such, the immediate candidates for full-measure saddle points over a polytope are the vertices, which are the only points in any polytope to have d-dimensional normal cones [74]. Any vertex is guaranteed to have a d-dimensional region of optimality, and so the focus from here onward will be on vertices as possible full-measure saddle points.

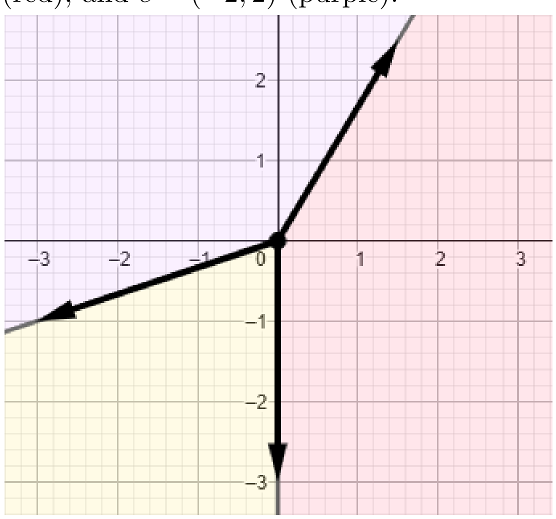
Example 6.2.5. In two dimensions, we can already identify polytopes which have full-measure saddle points. Consider a triangle with vertices at $a = (0, -1)$, $b = (3, -1)$,



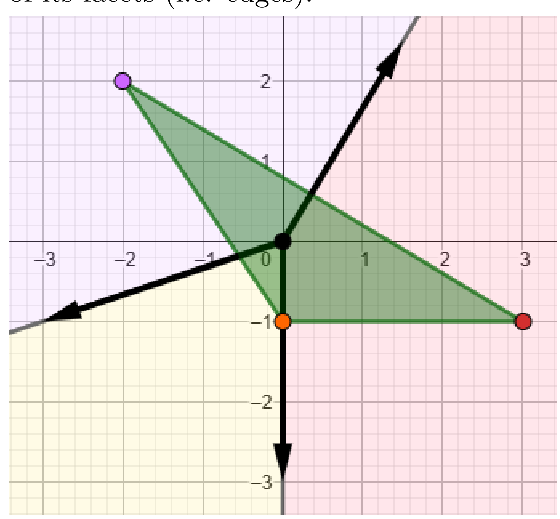
(a) A green-shaded triangle with color-coded vertices $a = (0, -1)$ (orange), $b = (3, -1)$ (red), and $c = (-2, 2)$ (purple).



(b) The triangle depicted in Figure 6.3a, overlaid with the 3 vectors normal to each of its facets (i.e. edges).



(c) The normal fan of the triangle depicted in Figure 6.3a, consisting of 3 regions subdividing $\mathbb{R}^2$. The regions are shaded orange, red, and purple.



(d) The normal fan overlaid with its corresponding triangle. The color of each vertex aligns with the color of one region of the fan, which is its normal cone.

Figure 6.3: A triangle constructed such that the vertex $a = (0, -1)$ is a full-measure saddle point. In Figure 6.3d, the vertex visibly resides on the boundary between its own normal cone and the neighboring normal cone of $b = (3, -1)$. For completeness, the normal vectors for each facet and the full normal fan are represented in Figure 6.3b and Figure 6.3c.

and $c = (-2, 2)$, as shown in Figure 6.3a. We can explicitly write out the three bounding hyperplanes that define this triangle, each of which is an inequality in $\mathbb{R}^2$:

$$\begin{aligned} H_1 :& x_2 \geq -1 \\ H_2 :& 3x_1 + 2x_2 \geq -2 \\ H_3 :& -3x_1 - 5x_2 \geq -4 \end{aligned}$$

Each of these hyperplanes has an associated normal direction that points outward from the interior of the polytope. These three directions are depicted as vectors in Figure 6.3b and are, respectively:

$$v_1 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} -3 \\ -1 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

These three directions can be used to partition $\mathbb{R}^2$ into three regions as shown in Figure 6.3c. Collectively, these comprise the normal fan of the triangle, and each region is the normal cone of one of the vertices; this is represented by the shading of the regions, which can be seen in Figure 6.3d to match the colors of the vertices.

Of particular interest here is the vertex $a = (0, -1)$, which in Figure 6.3d clearly sits on the boundary between its own normal cone (shaded orange) and the normal cone of the vertex $b = (3, -1)$ (shaded red). By the definition of the normal cone (Definition 2.1.8), any point x in the orange region obeys $x \cdot a \geq x \cdot y$ for all other points y in the triangle. Similarly, any point x in the red region obeys $x \cdot b \geq x \cdot y$ for all other points y in the triangle. Since a sits on the boundary of these two regions, for any $\varepsilon > 0$, the ball $\mathcal{B}_\varepsilon(a) \subset \mathbb{R}^2$ will contain some portion A solely in the normal cone of a, and some portion R solely in the normal cone of vertex b (both excluding the boundary). Applying $T(x)$ to any point $x_0 \in A$ will immediately yield $T(x_0) = a$, and applying $T(x)$ to any point $x_0 \in R$ will similarly yield $T(x_0) = b$. Therefore, a is a full-measure saddle point.

Based on these observations, we can develop a complete characterization of the conditions required for vertices to be full-measure saddle points. Note that by "full-dimensional" we mean d-dimensional.

Lemma 6.2.6. *Let v be a vertex of some polytope $P \subset \mathbb{R}^d$. Then v is a full-measure saddle point with respect to $T(x)$ if and only if v lies on the boundary between its own region of optimality C_v and the region of optimality C_w for some other vertex $w \neq v$, where both $P \cap C_v$ and $P \cap C_w$ are full-dimensional.*

Proof. The reverse direction follows directly from definitions. If $P \cap C_v$ is full-dimensional, then its intersection with any arbitrarily small neighborhood of v must also be full-dimensional. Moreover, any point in $P \cap C_v$ will be attracted to v, so the intersection of $P \cap C_v$ with this small neighborhood must be contained in A. Therefore A must be full-dimensional. A similar argument proceeds for R, using $P \cap C_w$, and so A and R are both full-dimensional, making v a full-measure saddle point.

We now prove the forward direction. Suppose v is a full-measure saddle point. Then for any arbitrarily small neighborhood of v within P, there exist two regions A, R which are full-dimensional and represent all points in the neighborhood that are attracted to or repelled from v, respectively. By definition, A must be contained in C_v, the region of optimality of v; since A is also contained in P, we have that $P \cap C_v$

is full-dimensional. Meanwhile, any points in R that are repelled from v must be attracted elsewhere. In particular, for them to be repelled from v, they cannot be in the normal cone of v, and will therefore be in the normal cone of at least one other vertex. Since R is full-dimensional and P has a finite number of vertices, there must be a vertex w such that $R \cap C_w$ is full-dimensional as well. However, R exists for any arbitrarily small neighborhood of v, which means C_w must include v. Since both C_v and C_w must contain v, it follows that v is on the boundary of these two regions, specifically on the boundary of its own normal cone and the normal cone of w. $\square$

Having established the existence and structure of full-measure saddle points, we now begin building towards a specific scenario where we can guarantee their appearance under a particular construction. Recall the discussion of normal cones and normal fans given in Section 2.1. For the next lemma, we note that a proper rotation is a bijective mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ such that the origin is preserved, Euclidean distances are preserved, and orientation is preserved.

Lemma 6.2.7. *Suppose two d-dimensional polytopes $P, P' \subset \mathbb{R}^d$ with normal fans F, F' respectively are related via proper rotation, i.e. there exists some rotation Φ such that $P' = \Phi(P)$. Then $F' = \Phi(F)$.*

Proof. Recall from Definition 2.1.11 that the normal fan of a polytope P can be understood as a method of portioning $\mathbb{R}^d$ into the normal cones of the polytope's vertices, which in turn include the normal cones of every face of the polytope. The normal cone of each facet of the polytope is simply its outward-directed normal vector, whereas the normal cone of each vertex is the cone generated by the vectors of the facets incident on that vertex (Lemma 2.1.9,Lemma 2.1.10). As such, the outward-directed normal vectors of the facets fully characterize the normal fan. It suffices to show that the normal vector of any facet of P' is equivalent to the rotation of the normal vector of the corresponding facet of P, as this ensures the vectors which characterize F precisely yield the vectors which characterize F'.

Note that since $P' = \Phi(P)$, each facet $A' \subset P'$ must have exactly one corresponding facet $A \subset P$ such that $A' = \Phi(A)$. Let α be the outward-directed normal vector of A; that is, not only is α normal to A, but for any point $p \in A$ and $\varepsilon > 0$, $p + \varepsilon\alpha \notin P$. Since Φ preserves orientation, $\Phi(\alpha)$ must be normal to $\Phi(A)$, and we can further note that $\Phi(p) \in A'$. Then $\Phi(p) + \varepsilon\Phi(\alpha) = \Phi(p + \varepsilon\alpha)$ cannot be in P' because $p + \varepsilon\alpha$ is not in P, verifying that $\Phi(\alpha)$ is the outward-directed normal vector of A'. This will hold for any facet, meaning all normal vectors carry over by rotation, and therefore $F' = \Phi(F)$. □

Lemma 6.2.8. *Given an $(d-1)$-dimensional polytope $P \subset \mathbb{R}^n$ with normal fan F, any orthogonal prism $Q = P \times I$ for some line segment I normal to P will have a normal fan characterized by the vectors of F in addition to the pair of normal vectors to the affine space containing P.*

Proof. Lemma 7.7 of Ziegler's *Lectures on Polytopes* states that the normal fan of the product of two polytopes $P \in \mathbb{R}^p$, $Q \in \mathbb{R}^q$ is equal to the direct sum of the normal fans of those polytopes [74]. The normal fan of any line segment considered as a one-dimensional polytope consists of two vectors directed parallel and antiparallel to the line segment, and the normal fan of P is fully characterized by the vectors of its facets, all of which are orthogonal to the line segment I by construction. Therefore the normal fan of the resulting d-dimensional prism Q will be the collection of cones generated by these vectors. □

Our interest now is in the connection between full-measure saddle points of a

$(d-1)$-dimensional polytope P embedded in $\mathbb{R}^d$ and full-measure saddle points of an associated orthogonal prism Q. We observe that any such polytope P must be contained in a $(d-1)$-dimensional affine space S, which must be parallel to some hyperplane $H \subset \mathbb{R}^d$ containing the origin. Moreover, there must be a rotation of $\mathbb{R}^d$ that bijectively takes H to the $x_1x_2...x_{d-1}$-hyperplane, which will also bring S normal to the x_d-axis. We will make use of the existence of such a transformation to simplify the following proof.

Proposition 6.2.9. *Let P be an $(d-1)$-dimensional polytope embedded in $\mathbb{R}^d$, with some vertex v that is a full-measure saddle point for $T(x)$ operating over P. Then for any orthogonal prism $Q \subset \mathbb{R}^d$ of P, at least one of the two vertices $u, u' \in Q$ that are constructed from v will be a full-measure saddle point for $T(x)$ operating over Q.*

Proof. By Lemma 6.2.7, normal fans rotate with their polytopes but are otherwise unchanged. Therefore, without loss of generality we can treat P as though its corresponding affine space S has the normal direction $\langle 0, 0, ..., 0, 1\rangle$, since the rotation of $\mathbb{R}^d$ unique to the hyperplane H would make this true and can be reversed after the prism is constructed. This allows us to consider Q as $P \times I$ for some interval I on the x_d-axis. Through the prism operation, each vertex of P becomes an edge of Q; in particular, $v \in P$ becomes an edge V of Q, with two endpoints u, u' that are vertices of Q. Q has two facets G, G' which are copies of P; these facets are parallel to each other, G with outward-directed normal vector $\langle 0, 0, ..., 0, 1\rangle$ and G' with the opposite vector $\langle 0, 0, ..., 0, -1\rangle$, and the edge V passes between them. Both u and u' occupy the position in their respective P-facets G, G' that was previously occupied by v. Note that if we consider u as $u = (u_1, u_2, ..., u_n)$, then u' must be $(u_1, u_2, ..., u_n - \ell)$ where ℓ is the length of the line segment I such that $Q = P \times I$.

Since v is a full-measure saddle point in P, there exists some vertex $w \in P$ such that v lies on the boundary of C_v and C_w within the setting of S, and that $P \cap C_v, P \cap C_w$ are $(d-1)$-dimensional sets. Let W be the edge of Q, with endpoints z, z' in the same P-facets G, G' as u, u' respectively, resulting from the prism operation over P. By Lemma 6.2.8, the normal fan of P is augmented by a pair of new vectors to get the normal fan of Q. Since these vectors are parallel to each other but orthogonal to the fan of P, they generate two d-dimensional cones for each $(d-1)$-dimensional cone of attraction corresponding to vertices of P; for v, C_v has become the pair of cones $C_u, C_{u'}$, and similar for w. The full-measure saddle-point behavior of v and the components of the normal fan preserved from P mean that every point on the edge V must lie on the boundary between two normal cones as well: one cone corresponding to either u or u', and one cone corresponding to either z or z'.

Note that the entire portion of the normal fan of Q which originated with P is contained in the $x_1 x_2 ... x_{d-1}$-hyperplane. As such, the normal cones of u and u' must be separated by this hyperplane, and likewise the normal cones of z and z'. Since V is an edge running in the normal direction to the bounding hyperplane, at least one of u or u' must not be contained in that hyperplane. Then either u, u' are both above or below the hyperplane, or u is above and u' is below. In the case that u, u' are both above the hyperplane, u is in its own normal cone, while also being on the boundary of the normal cone of z, meaning it will be a full-measure saddle point. In the case that u, u' are both below the hyperplane, u' is in its own normal cone and also in the normal cone of z', making it a full-measure saddle point. In the case that u is above and u' is below the hyperplane, both are in their own normal cones and in the normal cones of z and z' respectively, so both will be full-measure saddle points. $\square$

The final stage of this proof leads immediately to a more detailed result – not only will at least one of the vertices constructed from the full-measure saddle point be a full-measure saddle point in turn, but we can articulate the precise condition under which both vertices will be full-measure saddle points.

Corollary 6.2.10. *Let P be a $(d-1)$-dimensional polytope embedded in $\mathbb{R}^d$, with some vertex v that is a full-measure saddle point for $T(x)$ operating over P. Then for any orthogonal prism $Q \subset \mathbb{R}^d$ of P, the two vertices $u, u' \in Q$ that are constructed from v will both be full-measure saddle points for $T(x)$ operating over Q if and only if, for the hyperplane $H \subset \mathbb{R}^d$ parallel to the two P-facets of Q, there exists some $\varepsilon > 0$ such that $(H + \varepsilon) \cap Q \neq \varnothing$ and $(H - \varepsilon) \cap Q \neq \varnothing$.*

Proof. The condition of the existence of some $\varepsilon > 0$ such that $(H + \varepsilon) \cap Q \neq \varnothing$ and $(H - \varepsilon) \cap Q \neq \varnothing$ states that Q must fully cross the hyperplane in $\mathbb{R}^n$ that is parallel to its P-facets G, G'. In the previous proof, it was showed that when this is true, u and u' are both full-measure saddle points. Suppose this condition is not met, i.e. Q is fully on one side of the hyperplane or is incident on it.

If Q is fully on one side of the hyperplane, then u and u' must both be in either the normal cone of u or the normal cone of u', depending which side Q resides on. The vertex in whose normal cone they reside will be a full-measure saddle point, as shown before; however, the other vertex is not in its own normal cone and is therefore not a fixed point in this scenario. As such, it cannot be a full-measure saddle point.

If instead Q is incident on the hyperplane H, then one of its P-facets must be contained in the hyperplane, since the hyperplane is parallel to them. Suppose G

and by extension u are contained in H, in which case u' is the full-measure saddle point guaranteed by Proposition 6.2.9. Then u sits on the boundary of its normal cone and the normal cone of u'. Suppose u is in fact a full-measure saddle point; then there must exist some $\varepsilon > 0$ such that there is a d-dimensional set $A \subset \mathcal{B}_\varepsilon(u) \cap Q$ for which iterating with T from any $x_0 \in A$ will converge to u. Since A is d-dimensional, it must not be contained in H. Consider some $x_0 \in A \setminus H$. Since $x_0 \in Q$ and $x_0 \notin H$, x_0 cannot be in the normal cone of u, meaning T will not converge to u, yielding a contradiction. Therefore u cannot be a full-measure saddle point.

We conclude that the only condition under which u, u' will both be full-measure saddle points is when Q fully crosses the hyperplane H, i.e. when $(H + \varepsilon) \cap Q \neq \varnothing$ and $(H - \varepsilon) \cap Q \neq \varnothing$ for some $\varepsilon > 0$. $\square$

Proposition 6.2.9 and Corollary 6.2.10 indicate that full-measure saddle points exist for $T(x)$ and moreover they can be iteratively generated in polytopes of arbitrarily high dimension. An interesting aspect of these points is that $T(x)$ can converge to them even though they are not true attractive fixed points. The potential impact of this quirk on the overall behavior and accuracy of $T(x)$ remains an open question.

Chapter Seven

Experimental results

To determine the viability of using semidefinite programming with $T(x)$ (SDP-ILO) as a data clustering algorithm, we constructed a variety of simulations to test the performance of the algorithm. We primarily compared SDP-ILO with the commonly-used k-means algorithm. As part of this study, we specifically looked at the ways both algorithms made use of the parameter k. In k-means clustering, the algorithm takes k as a parameter and always produces a clustering with exactly k clusters; that is, the solution space of k-means is the collection of partitions of the data with exactly k non-empty blocks. The limitation of k-means to exactly k-block partitions is nonideal in real-world data clustering settings. Typically, there is little or no starting intuition on how many clusters exist in a dataset; thus, using k-means requires running it for many k-values to see a wide range of possible clusterings and identify the best one. By contrast, the solution space of SDP-ILO is the collection of partitions with *at most* k non-empty blocks, meaning theoretically the algorithm could end with a clustering that uses fewer than k blocks if this fits the data better than the provided parameter value k. One key question was whether this behavior would appear in practice; this capacity would greatly improve the viability of SDP-ILO in real data settings, since no other commonly-used clustering algorithms have the capacity to restrict to fewer than k blocks given an input of k.

7.1 Methods

In the context of data clustering, the SDP-ILO algorithm works as follows. An input of n data points is reinterpreted as the complete graph $K_n = (V, E)$, each point becoming a vertex, with edges weighted by the Euclidean distances between data points. The weights are stored as a distance matrix M, which is a symmetric n-by-n matrix with zeros along the diagonal. The standard k-max-cut objective, which maximizes the weights of edges crossing the cut, is relaxed to a semidefinite programming problem using the same unit vector approach as the Goemans-Williamson algorithm [31]:

$$\begin{aligned} \text{Maximize: } & \frac{k-1}{k}\sum_{i<j} w_{ij}(1 - v_i \cdot v_j) \\ \text{Subject to: } & v_i \in S_{n-1} \quad \forall i \in V \\ & v_i \cdot v_j \geq -\frac{1}{k-1} \quad \forall i \neq j \end{aligned}$$

The program is then solved directly using the input M as the collection of weights w_{ij}. The solution to the semidefinite program is a matrix X in the k-way elliptope $\mathcal{L}_{n,k}$ and is thus a symmetric positive semidefinite matrix with ones along the diagonal and all entries in the range $[-\frac{1}{k-1}, 1]$. Th output matrix X is used as the initial point x_0 for the iterative algorithm $T(x)$ operating over the domain $\mathcal{L}_{n,k}$. Once iteration of $T(x)$ reaches a stage such that x_{i+1} is within some user-selected small distance of x_i, it is deemed complete and the nearby fixed point is returned as the clustering solution.

To compare the performance of SDP-ILO to k-means, for each experimental dataset, both algorithms were used to generate a clustering of the data for each of several input k-values. One of these input values was always the correct number of clusters k_{true} used to produce the data, where k_{true} was set in the range $5 \leq k_{true} \leq 12$. The other inputs ranged from $k = 2$ to $k = 20$, to capture a range of behavior when given smaller or larger input values than k_{true}. The clusterings were scored using three different clustering assessment methods. The collections of each of the scores for each algorithm were plotted over the changing input values k. Examining the scores over k allowed a view of the trends in the scores, to see which inputs yield the best performance and how high the contrast is between different inputs of k. The expectation was that the algorithms should each score best with an input of $k = k_{true}$;

beyond this, however, a strong clustering algorithm should exhibit a large change between $k = k_{true}$ and $k \neq k_{true}$. Since clustering is often done in a setting where the true number of clusters is unknown, a striking and observable improvement in score at a particular input k is compelling evidence for the true number of clusters; therefore, an algorithm must possess this quality to be broadly usable.

7.1.1 Datasets

An important consideration of experimental design was "neatness", i.e. well-separatedness, of the input data. Neat data is in theory the easiest to cluster; however, when asking an algorithm to find some number of clusters k that does not match the natural number of clusters in the data, performance can vary significantly. On well-separated data, a good clustering algorithm should not only identify the natural clusterings when given $k = k_{true}$, but also provide a clear indication of which k-value is in fact the true one. Meanwhile, on messy data, it is not expected that an algorithm will produce a clustering that perfectly matches the original data groupings, but rather one which is locally similar to the original data groupings. In this setting, too, being able to produce clustering results that point to the correct k-value as k_{true} is crucial to performance. As such, a variety of experiments were performed, using datasets selected both for extreme well-separated niceness and for extreme messiness.

One set of experiments was run by generating data in $\mathbb{R}^2$, from a collection of k_{true} Gaussian distributions. A few example datasets generated in this fashion are given in Figure 7.1. The number of distributions used to generate the data varied from 4 through 8, with the majority of experiments conducted using 8 distributions. The centers of the distributions were evenly spaced around a unit circle, and the standard

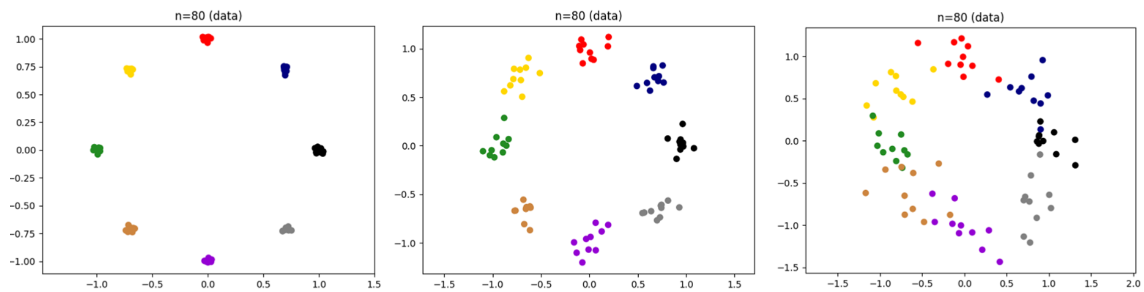


Figure 7.1: Three sample Gaussian datasets generated for clustering testing. Each has 8 evenly-spaced Gaussian source distributions, and 10 data points were sampled from each, yielding a total of 80 points in each dataset. The data is color-coded according to its source distribution. The first dataset implements a standard deviation of 0.02, the second uses a standard deviation of 0.1, and the third has a standard deviation of 0.25 for each of the 8 Gaussian distributions.

deviations varied from 0.02 to 0.3 to test the algorithm on both well-separated data and messy data. Using a standard deviation of 0.1 or less with up to 8 distributions consistently yielded visually separable data, whereas increasing the standard deviation to 0.15 or greater for 8 distributions yielded data that could not be cleanly separated by a human observer. For these experiments, the same number of data points were generated from each distribution, totaling no more than 100 points per dataset.

Another set of experiments focused on the MNIST dataset, a collection of 28-by-28-pixel grayscale images of handwritten digits [48]. The images have been scaled and centered to introduce some consistency for comparison, but there is a tremendous amount of variety in the appearance of each digit depending on the writer. A sample of these images, which demonstrates this variation, is given in Figure 7.2. For each experiment using this data, images were selected at random, without standardizing how many images of each digit type would be included. This allowed study of the performance of SDP-ILO when the true clusters were not equal in size. The true number of clusters was considered to be $k_{true} = 10$. The data was fed to SDP-ILO and k-means for clustering in two forms. One approach was to use the raw

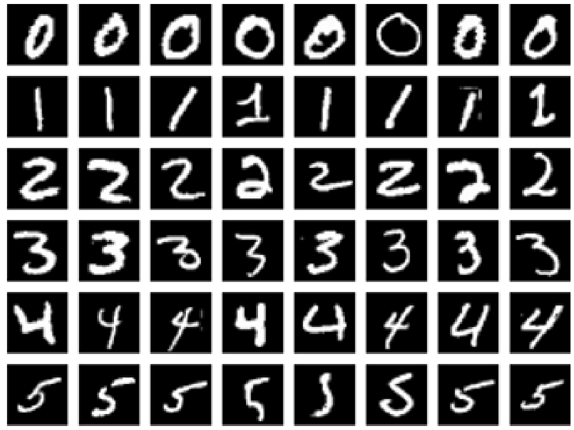


Figure 7.2: Sample images from the MNIST dataset, representing the digits 0 through 5. The dataset also includes samples of digits 6 through 9. From these images, it can be seen that within a category of digits, there is immense variation in the images produced by handwriting those digits. (Wikipedia: MNIST)

image data, consisting of 784-dimensional vectors of pixel values; since the data was so high-dimensional, only 50-60 images were sampled for each experiment. The other approach was to preprocess the image data using TensorFlow, which for each individual image returned a 10-dimensional vector representing the probabilities of the image belonging to each digit class. This method mimicked the work of Mixon et al. in [51] and was implemented explicitly on the image subset used by Mixon, sampled in batches of 200, to compare performance not only with k-means but also with the Mixon algorithm.

7.1.2 Scoring methods

Three scoring methods were used to compare the performance of SDP-ILO and k-means in each experiment. Two of these methods relied on knowledge of the true classifications of each data point, i.e. which Gaussian distribution or digit class the data point originated from. These were the adjusted Rand score and the adjusted mutual index, both of which were also utilized in [51]. The third method, the Davies-

Bouldin score, does not account for the true source of the data, since in many real clustering settings these true distributions are unknown and cannot be considered. All three methods were implemented using the scikit-learn package in Python [54].

The adjusted Rand score has a range of $[-0.5, 1]$, with a score of 1 indicating perfect agreement between a clustering and the original data clusters. A random labeling of the data points will give a score near zero. No assumptions are made on the shapes of the clusters, and the scoring is symmetric – the Rand score will be the same regardless of which of its two clustering inputs is identified as the ground truth. The original Rand score operates by considering the $\binom{n}{2}$ pairs of objects in a collection of size $[n]$, counting up all of the pairs for which the original clusters and a computed clustering agree on whether those pairs should be together or separated, and dividing the total by $\binom{n}{2}$ [39]. The adjusted Rand index as implemented in scikit-learn is designed to account for the influence of chance; that is, if a clustering was built by randomly generating cluster assignments for each data point, it would agree with some fraction of the original cluster information by pure luck. To ensure that a randomly-generated clustering is considered the minimally-desirable outcome and should receive a score of 0, the adjusted Rand score computes the "expected Rand score" – the average Rand score of a randomly clustering – and scales for it [39]:

$$AdjustedRand = \frac{Rand - ExpectedRand}{MaxPossibleRand - ExpectedRand}$$

By utilizing this scaling, the adjusted Rand score ensures that the full range of values in the interval $[0, 1]$ is meaningful: a score near 0 indicates that a computed clustering is no better than random assignment. This metric is widely utilized but is most reliable when handling evenly-distributed data from similar distributions, and is thus especially well-suited to the Gaussian data samples in this study. Since it is broadly implemented in the literature as a basic universal similarity measure for clusterings, it is utilized here across all experiments for direct comparison.

Adjusted Mutual Information (AMI) is similar to the adjusted Rand score in that a score of 1 indicates perfect agreement with the true clusters, and a score of 0 indicates that the clustering output and the ground truth are independent of each other, i.e. the metric is normalized against chance. Moreover, this measure is also similarly symmetric. However, the two scores are in fact prioritizing two different approaches to similarity. The Rand score utilizes a straightforward pair-counting approach, while the Adjusted Mutual Information score has an information-theoretic approach to clustering evaluation [58]. Without delving too far into the details of information theory, "mutual information" uses entropy to compute a notion of how much information is identical between two different clusterings; the idea is that if the clusterings are similar to each other, then each clustering should serve as a good predictor of the other's behavior [69]. Like the Rand score, the initial computation of mutual information is not normalized, and can be adjusted by incorporating the expected mutual information score of a randomly-generated clustering [69]:

$$AdjustedMI = \frac{MI - ExpectedMI}{\max(ClusteringEntropies) - ExpectedMI}$$

Unlike the adjusted Rand score, this index can be directly interpreted as a distance measure between two clusterings; the Rand score is a good similarity measure but is not a mathematically-rigorous metric [69]. This makes AMI more robust for data that is unevenly sampled from very different-looking distributions, as is the case with the MNIST samples utilized in this study.

The Davies-Bouldin index is used to score a clustering for which the ground truth is not known. Although the ground truth was available for every experiment, it would not typically be available in a real-world setting, as the purpose of clustering is often to identify and ultimately to characterize previously-unknown clusters within a large dataset. Therefore, it is also important to assess whether clusterings which appear strong in comparison to a known ground truth will still appear strong when assessed independently of that truth. The Davies-Bouldin index accomplishes this by comparing the distances between clusters to the sizes of those clusters, leading to a measurement system very different from the other two metrics: in this scoring structure, the lowest possible score is 0, and scores near zero indicate more effective, well-separated partitioning of the data. The Davies-Bouldin score was explicitly designed with the intention that across clusterings computed with different numbers of clusters, the correct number of clusters would minimize the score [20]. It accomplishes this using the following formula, where $i \neq j$ represent two different clusters in some clustering with N clusters in total [20]:

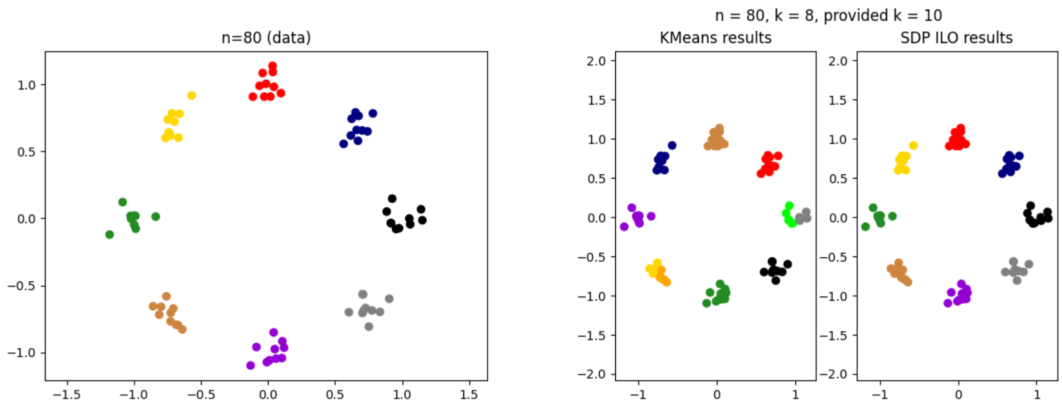
$$DB = \frac{1}{N} \sum_{i=1}^{N} \max_{i \neq j} \left(\frac{S_i + S_j}{M_{ij}} \right)$$

S_i and S_j are the dispersions of the clusters represented by i and j, i.e. each is the sum of squared distances between all pairs of points in the corresponding cluster. M_{ij} represents distance between clusters i and j, computed as the sum of squared distances between all pairs of points across both clusters. With this construction, seeking the smallest possible value of DB is approximately equivalent to seeking the clustering according to which the clusters are most geometrically well-separated from each other in space [20]. The Davies-Bouldin score is thus designed to act as a mathematical formalization of the visual characteristics by which clusters can be identified and assessed in two or three dimensions, making it a particularly intuitive metric so long as the data under consideration exists in Euclidean space, as is the case for both the Gaussian datasets and the MNIST samples.

7.2 Results

7.2.1 Gaussian data tests

A few trends emerged across the experiments with the Gaussian data. SDP-ILO was consistently able to identify the true clustering whenever using the input $k = k_{true}$ on data with small (≤ 0.1) standard deviations, i.e. data that was visually separable. This matched the performance of k-means for neat data. For $k < k_{true}$, the true clustering was not in the solution space for SDP-ILO or for k-means. In these cases, SDP-ILO regularly received lower scores than k-means in the Rand and AMI metrics, and higher scores in the Davies-Bouldin metric. Both of these outcomes represent less effective clusterings of the data for the incorrect k-values. SDP-ILO also consistently produced exactly k clusters for its input value k in these trials. For $k > k_{true}$, the true clustering was in the solution space for SDP-ILO, but not for k-means. In the majority of trials, even with this inclusion, SDP-ILO still produced exactly k clusters. However, when using k near k_{true}, it occasionally instead returned a clustering with fewer than k clusters, while never returning fewer than k_{true} clusters. For one specific case, using $k_{true} = 8$ and $k = 10$ on a set of distributions with a standard deviation of 0.08, SDP-ILO reproduced the correct set of 8 clusters, as can be seen in Figure 7.3. In this trial, the SDP-ILO clusterings for $k = k_{true} = 8$ and $k = 10$ were identical, so they had identical scores in each of the clustering metrics.

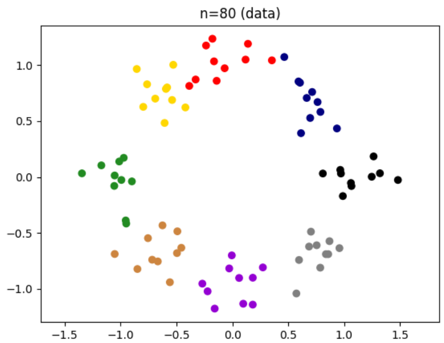


(a) The original data, colored by source distribution

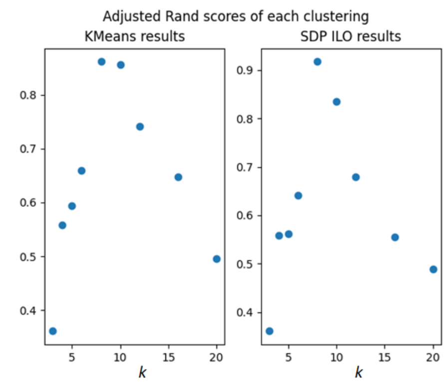
(b) The clusterings given by k-means and SDP-ILO, colored by assigned cluster

Figure 7.3: The data consists of 10 data points selected from each of 8 evenly-spaced Gaussian distributions, each with a standard deviation of $s = 0.08$. Using k-means with $k = 10$ produced 10 clusters, splitting the data groupings colored brown and black in Figure 7.3a into two smaller clusters each. Using SDP-ILO with $k = 10$ produced 8 clusters, exactly matching the original data sources.

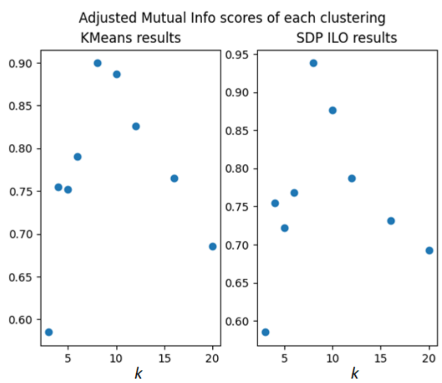
When SDP-ILO did not re-identify the true clustering and did not reduce the number of clusters it returned, it once again scored worse than k-means in all three metrics. For the Rand score and AMI, this produced plots with a sharp peak at $k = k_{true}$ for both k-means and SDP-ILO, with SDP-ILO having the more severe change. For the Davies-Bouldin score, the plots had a similar structure, but with a valley rather than a peak since a lower score indicated a better clustering. As standard deviation was increased, making the data less separable, both k-means and SDP-ILO yielded gradually less pronounced peaks and valleys due to the messiness obscuring the source distributions. Figure 7.4 shows these trends for a standard deviation of $s = 0.15$, high enough for the true distribution sources to be difficult to identify visually but low enough for effective clustering.



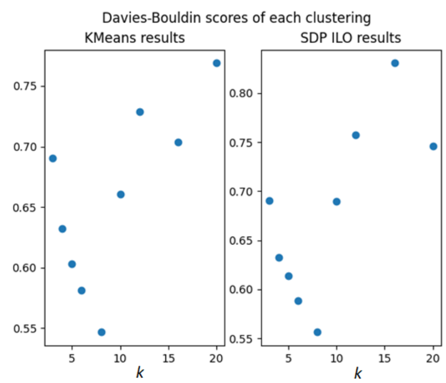
(a) Data from 8 Gaussian distributions with standard deviations of $s = 0.15$



(b) Adjusted Rand Index scores over k for both algorithms



(c) Adjusted Mutual Information scores over k for both algorithms



(d) Davies-Bouldin scores over k for both algorithms

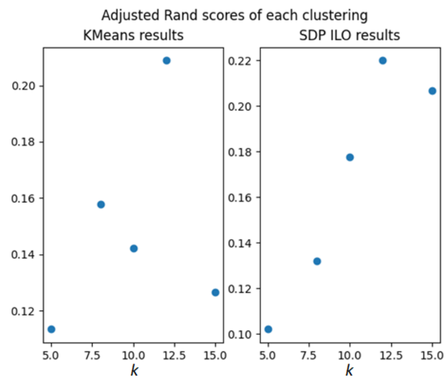
Figure 7.4: In the experiments run with the data shown in Figure 7.4a, SDP-ILO achieved better scores for $k = k_{true} = 8$ than k-means in the Adjusted Rand and Adjusted Mutual Information measures, but a slightly worse score in the Davies-Bouldin metric. For $k \neq k_{true}$, especially $k > k_{true}$, SDP-ILO exhibits notably worse scores than k-means in all three metrics.

Some computational obstacles arose during these tests. The decision to keep dataset sizes low in the Gaussian tests, a decision which carried through to the MNIST tests, was largely driven by a rapid growth in runtime as n increased. Improvements in programming implementation may mitigate the runtime issue and allow testing on larger datasets in the future. In addition, one trial resulted in a computational anomaly. The iterative algorithm was caught in a six-point loop without getting close enough to a fixed point to terminate. After several iterations around this loop, the process had to be terminated manually. This phenomenon did not reoccur and is believed to be an artifact of rounding error, as it defies the known properties of the iterative algorithm.

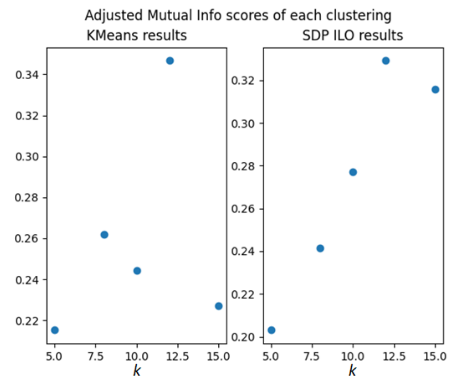
7.2.2 MNIST data tests

Both k-means and SDP-ILO performed significantly worse on MNIST data than on Gaussian data. As shown in Figure 7.5, not only did both algorithms earn poor scores in all three metrics regardless of the input value for k, but neither algorithm was able to consistently obtain an improved result for $k = k_{true} = 10$ for any of the trials on the raw data. It can be surmised that raw MNIST data is too messy for either algorithm to reliably cluster.

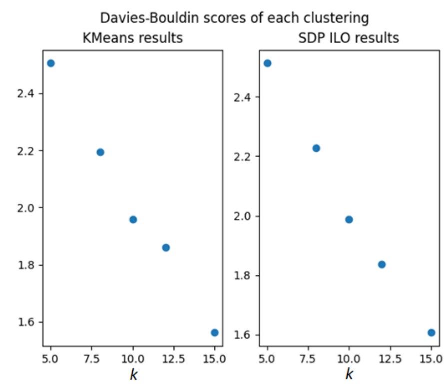
The results observed for MNIST data preprocessed with TensorFlow were more interesting. With the incorporation of this step, both k-means and SDP-ILO were able to recover significantly with regards to clustering scores. Instead of a constant improvement in cluster scoring as k increased past k_{true}, k-means demonstrated decreased or negligible improvements for $k > k_{true}$ in all three metrics, presenting an alternative mechanism by which k_{true} might be identified. However, SDP-ILO returned fully to the pattern observed in the Gaussian data, where its scores achieved a peak at $k = k_{true}$ and then regressed. This regression was occasionally overridden by a further improvement for especially high k, particularly in Davies-Bouldin score, indicating that the messiness of the data can still overwhelm the algorithm when the true number of clusters is unknown.



(a) Rand scores over k for raw MNIST data

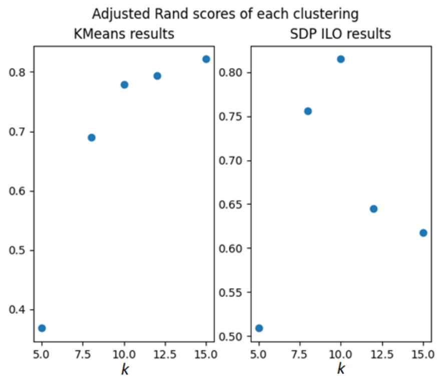


(b) AMI scores over k for raw MNIST data

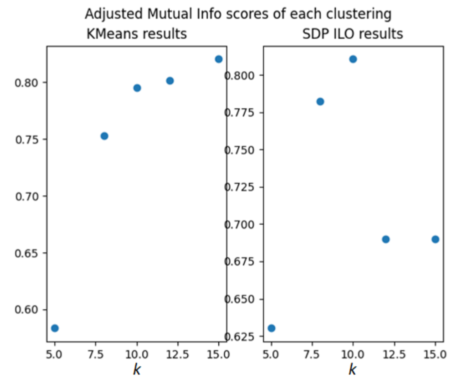


(c) Davies-Bouldin scores over k for raw MNIST data

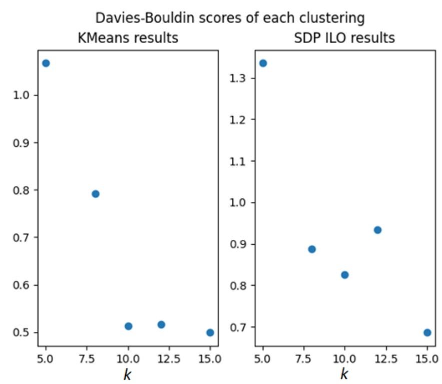
Figure 7.5: On raw samples of MNIST datapoints, neither k-means nor SDP-ILO could reliably yield a strategy for identifying $k_{true} = 10$, and all scores were poor with any input k.



(a) Rand scores over k for preprocessed MNIST data



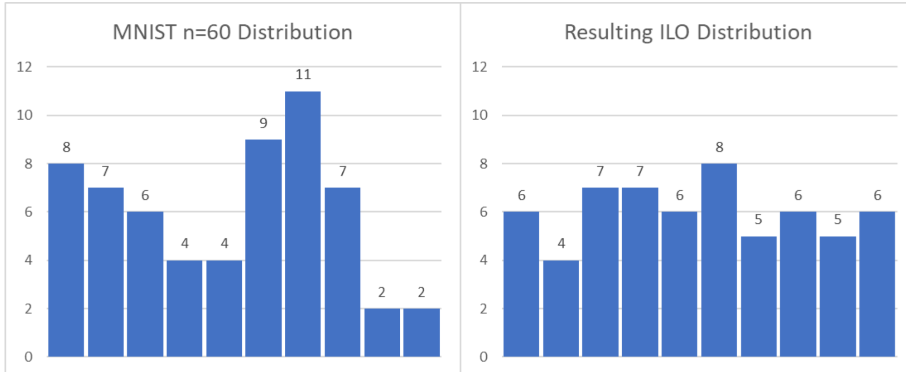
(b) AMI scores over k for preprocessed MNIST data



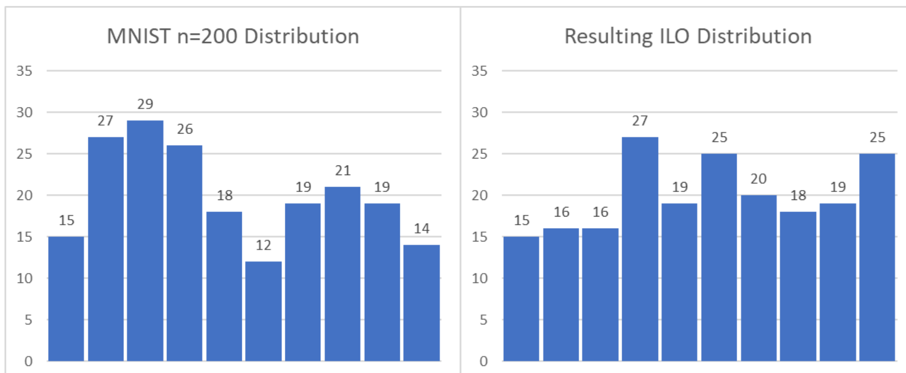
(c) Davies-Bouldin scores over k for preprocessed MNIST data

Figure 7.6: When using preprocessed data, the scores for both algorithms improved substantially in comparison to the raw MNIST data used in Figure 7.5, and SDP-ILO began exhibiting sharp peaks at $k = k_{true}$ in the Rand and AMI scores.

In addition to generating the clustering scores, histograms were generated to track cluster sizes produced by SDP-ILO for comparison to the true cluster sizes of each raw MNIST sample. The number of samples to be taken from each digit type was not prespecified, allowing high variation to occur in cluster sizes. In general, SDP-ILO produced clusters that were more even in size than the true clusters from the data. This trend persisted for the preprocessed data tests as well, although typically the true cluster sizes were more even in these tests due to the higher sample size.



(a) Original (left) and SDP-ILO (right) cluster sizes for an MNIST sample with $n = 60$.



(b) Original (left) and SDP-ILO (right) cluster sizes for an MNIST sample with $n = 200$.

Figure 7.7: In general, the clusters constructed by SDP-ILO evened out the sizes of the true clusters. This occurred for both raw and preprocessed MNIST data.

Due to the visual nature of the data, clustering performance can also be informally assessed according to how well the clusters for $k = k_{true} = 10$ visibly represent the clusters that are known to exist in the dataset. To account for this perspective, during experiments using preprocessed data with $n = 200$, the datapoints placed in each cluster by SDP-ILO for $k = 10$ were averaged together to produce a mean representative image for each cluster, such as the ones shown in Figure 7.8. When converted to heatmap images, the collection of 10 representative images of the SDP-ILO clusters consistently indicated that each cluster had a dominant digit type within it, and no two clusters had the same dominant digit.

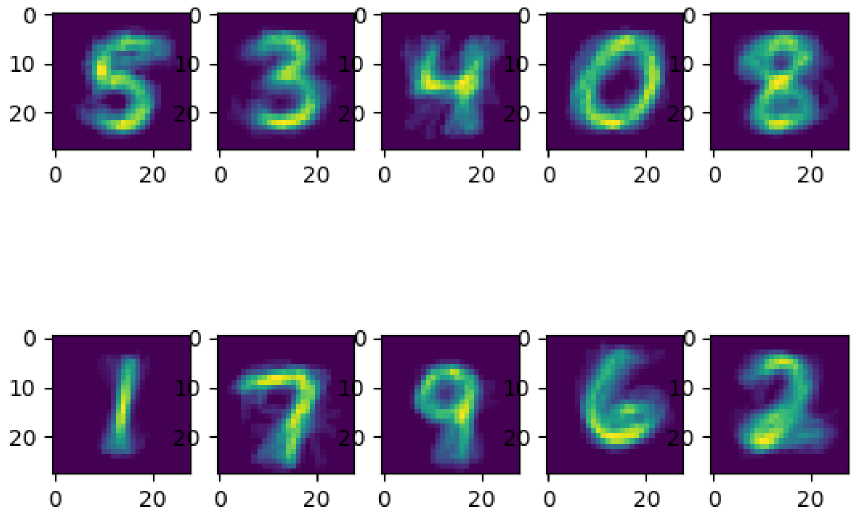


Figure 7.8: Each of the 10 clusters constructed by SDP-ILO, when averaged together, produced an image recognizable as a unique digit. Top row, left to right: 5, 3, 4, 0, 8. Bottom row, left to right: 1, 7, 9, 6, 2.

Comparison to the Mixon algorithm was solely undertaken using $k = 10$ and examining adjusted Rand score and AMI, since these were the two measures used in the Mixon study [51]. Out of 10 experiments done with samples of $n = 200$ from the preprocessed data, SDP-ILO always either scored worse in both metrics or better in both metrics, and it scored better on 3 out of 10 tests. On all tests, the scores in both metrics were extremely close to each other, as can be seen in Figure 7.9. Moreover, both algorithms consistently performed better according to the adjusted mutual information score than the adjusted Rand index.

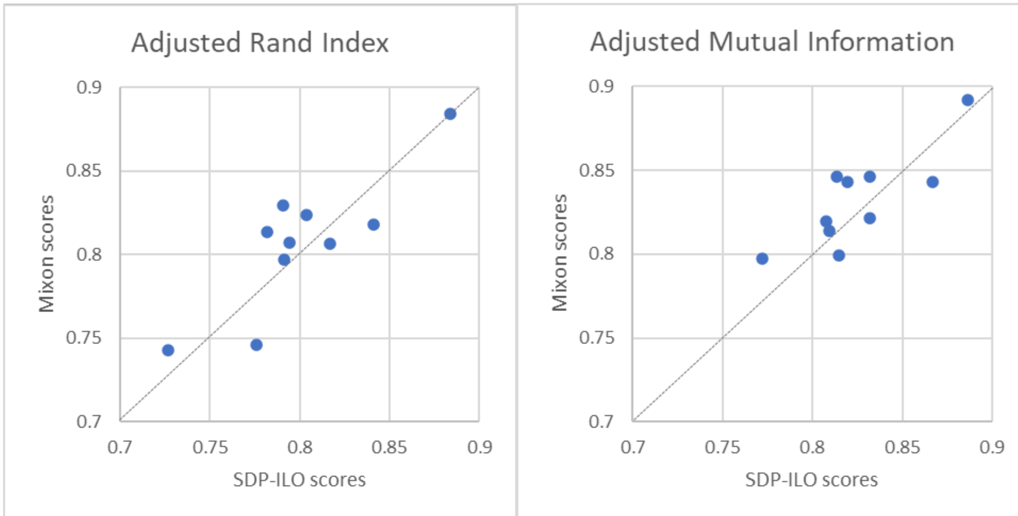


Figure 7.9: Clustering scores by the Mixon algorithm vs. SDP-ILO in both the adjusted Rand index (left) and adjusted mutual information (right) measures. The dotted line on each chart represents equivalent scores. Dots above the line indicate a higher score by the Mixon algorithm. The three cases where SDP-ILO performed better than Mixon in the adjusted Rand score are the same three experiments in which SDP-ILO performed better than Mixon in the adjusted mutual information score.

7.3 Discussion

From the Gaussian experiments it can first and foremost be observed that the primary theoretical advantage of SDP-ILO is in fact computationally feasible. In multiple tests with $k > k_{true}$, SDP-ILO returned fewer than k clusters, indicating that k is a number that does not fit the data and – crucially – that a lower number of clusters is more suitable. This happened solely with standard deviations of 0.1 or less, indicating that well-separated data is more likely to draw out this behavior; there is an intuitive logic to this, as constructing more clusters than truly exist in a well-separated dataset requires splitting up a cluster that is tightly-packed in comparison to the overall range of the data spread.

However, even for small standard deviations, SDP-ILO did not exhibit this behavior consistently. It is unclear why only certain randomly-generated datasets provoked SDP-ILO to return fewer than k clusters. One possible contributing factor might be the size of the solution space in which SDP-ILO is navigating in this context. The number of extremal points of $\mathcal{L}_{n,k}$ is equal to the number of partitions of n with at most k non-empty blocks, which as established in the first part of this thesis is a sum of Stirling numbers $\left\{ {n \atop i} \right\}$ of the second kind for $1 \leq i \leq k$. In the case of this simulated data (and most real datasets), k is significantly smaller than n, and certainly less than half of n, meaning that $\left\{ {n \atop k} \right\}$ will be much larger than $\left\{ {n \atop k-1} \right\}$. With such an overwhelming fraction of attractive fixed points being exact k-block partitions, it is conceivable that the proportion of $\mathcal{L}_{n,k}$ contained in the region of attraction for one of these points is simply too large for SDP-ILO to reliably evade them during calculations. The outcome of the semidefinite programming step, which gives the starting point for the iterations, is likely crucial to whether the iterations converge to a k-block clustering.

In addition to demonstrating its capacity to reduce the number of returned clusters in practice, SDP-ILO also provided consistently sharper indications of the correct k-value than k-means across almost all Gaussian datasets (barring some extremely messy data) as well as the preprocessed MNIST samples, for all three metrics used to assess the clusterings. SDP-ILO consistently produced clusterings that had worse scores than those of k-means whenever $k \neq k_{true}$, but the practical impact of this is that the charts tracking clustering scores over k show greater clustering improvement at $k = k_{true}$ when using SDP-ILO. The significance of this is especially relevant when looking at these charts for the MNIST experiments as well as the Gaussian samples with very high standard deviations. MNIST is a very messy dataset, and this is reflected in how even after cleaning and processing this data, k-means often still shows improvements or a steadying in clustering score as k increases beyond $k_{true} = 10$, as in Figure 7.6. These improvements decrease significantly for $k > k_{true}$, which can be used to hypothesize the value of k_{true} but is not as clear of an indicator as the sharp peaks observed in the Gaussian data. By contrast, SDP-ILO continues to exhibit a regression in clustering score after $k = k_{true}$ in the Rand index and MI score, giving a much stronger indication of k_{true}. Unfortunately, the Davies-Bouldin score is a less regular indicator, with SDP-ILO scoring worse than k-means on $k = k_{true}$ and not displaying a consistent trend in score for higher values of k. Further investigation is needed to analyze SDP-ILO trends in this scoring system for messy data and determine a strategy for reliably identifying k_{true}.

At present, the algorithm appears to display a preference for evenly-balanced clusterings; however, this may be due to the objective function originally utilized during implementation, rather than a bias within the algorithmic process. The objective function for the k-max-cut problem directly incorporates the sizes of the graph components resulting from a cut; this may be introducing a preference for a more uniform distribution of cluster sizes. Many objective functions used for clustering do not utilize any such element. The slightly weaker performance of SDP-ILO as a clustering algorithm in comparison to Mixon may also be a result of this unusual characteristic of the original max-cut design. Changing the objective function will alter the optimal outcome, but the extent of this alteration cannot be determined without additional investigation.

7.4 Experimental Future Directions

Ultimately, the long-term goal of these studies and similar investigations is to determine the approximation constant of the SDP-ILO algorithm and see how it compares to other semidefinite programming approaches to clustering and max-cut. Further investigation is especially warranted with regards to the apparent bias against uneven cluster sizes; this should begin by examining to what extent this behavior changes when the max-cut objective function is replaced with a more traditional clustering function when implementing the algorithm. To dig into this and other less-regularized data settings, more experiments should be conducted in the Gaussian simulation space. Thus far, the Gaussian simulations have varied the number of distributions, their standard deviations, and the number of data points generated per distribution. This could be expanded upon by varying the spacing of the distributions, allowing different numbers of points to be generated from each distribution, and allowing distributions to have different standard deviations.

Another line of inquiry that should be pursued with this algorithm will be whether there is some threshold for well-partitioned data at which the algorithm will be guaranteed to return the true clustering. Based on the observations made with these experiments, we have the following conjecture on data niceness:

Conjecture 7.4.1. *Suppose in a dataset with k_{true} clusters, the minimal distance between any two points in different clusters is greater than the maximal distance between any two points in the same cluster. Then given the input $k = k_{true}$, the algorithm SDP-ILO will return the true clustering of the dataset.*

This conjecture was not violated in any observed experiment. However, it has not yet been assessed analytically; one possible adjustment would be that a scaling factor may be necessary to prove this statement. Additional work towards this or any other conjecture regarding the behavior of SDP-ILO on well-separated data would improve understanding of the algorithm's capacity.

Finally, k-means is not the only commonly-used clustering algorithm, and in fact it differs substantially from SDP-ILO in how it seeks an optimal clustering. Some work, as in [51], has been done to apply semidefinite programming to clustering; direct comparisons between SDP-ILO and these algorithms, using similar approaches to those outlined here, will provide valuable insight to the long-term viability of these methods and which refinements will maximize their utility. In particular, a rigorous study should be undertaken on the effects of shifting the objective function utilized by SDP-ILO to a version more specialized to clustering, and a fresh comparison to other clustering algorithms including Mixon should be undertaken if this alteration appears beneficial for the application.

Part III

On Parking-Function Polytopes

Chapter Eight

Generalized Parking-Function Polytopes

The results presented in this chapter were originally published on arXiv as [8] and have been reprinted here in full, with some added setup and edits for clarity in the context of this document. The project was conducted in collaboration with Margaret Bayer, Steffen Borgwardt, Spencer Daugherty, Aleyah Dawkins, Danai Deligeorgaki, Hsin-Chieh Liao, Tyrrell McAllister, Angela Morrison, Garrett Nelson, and Andrés Vindas-Meléndez, beginning at the Graduate Research Workshop in Combinatorics hosted by the University of Wyoming in 2023. All notation in this chapter is specific to the domain of parking functions. The polytope background presented in Section 2.1 applies to the polytopes discussed here, but their structure and behavior should otherwise be considered from a fresh perspective.

8.1 Classical and Generalized Parking Functions

We introduce classical parking functions through their traditional motivating scenario. Consider a line of n cars entering a one-way street, one at a time, and attempting to park in one of the n spots along the street. Each car i has a preference a_i for which spot to park in, and will begin by driving directly to their preferred spot. If the preferred spot is empty, the car will park. If the preferred spot is occupied, the car will progress down the street until the next open spot, in which it will park. If there are no more open spots before the end of the street, the car will fail to park. A *parking function* is a sequence of spot preferences $(\alpha_1, \alpha_2, ..., \alpha_n)$ such that all n cars will successfully park. This can be more rigorously codified as follows [62]:

Definition 8.1.1. A **parking function** is a sequence $(\alpha_1, ..., \alpha_n)$ of positive integers such that their nondecreasing rearrangement $\alpha'_1 \leq \alpha'_2 \leq ... \leq \alpha'_n$ satisfies the condition

$$\alpha'_i \leq i \quad \forall i \in [n]$$

Example 8.1.2. Consider $n = 3$. It can be checked against the definition that $(1, 1, 2)$ and $(2, 1, 2)$ are parking functions, while $(2, 2, 2)$ is not. Examining these from the perspective of the motivating scenario, we find the following outcomes:

- $(1, 1, 2)$: The first car prefers spot 1. On arrival, spot 1 is empty, so it parks. The second car also prefers spot 1, but when it arrives, spot 1 is occupied. It proceeds forward. The next open parking spot is spot 2, so the second car parks in spot 2. The third car prefers spot 2, but when it arrives, spot 2 is occupied. Proceeding forward, it finds spot 3 empty and parks there. All three cars have parked successfully, making $(1, 1, 2)$ a parking function.

- $(2, 1, 2)$: The first car prefers spot 2. On arrival, spot 2 is empty, so it parks there. The second car prefers spot 1. On arrival, spot 1 is empty, so it parks in spot 1. The third car prefers spot 2, but when it arrives, spot 2 is occupied by the first car. Proceeding forward, it finds spot 3 empty and parks there. All three cars have parked successfully, making $(2, 1, 2)$ a parking function.

- $(2, 2, 2)$: The first car prefers spot 2. On arrival, spot 2 is empty, so it parks there. The second car also prefers spot 2, but when it arrives, spot 2 is occupied. It proceeds forward. The next open parking spot is spot 3, so the second car parks in spot 3. The third car also prefers spot 2, but when it arrives, spot 2 is occupied. It proceeds forward and finds spot 3 occupied as well. The third car is thus unable to park and leaves. Spot 1 is left unfilled, because none of the cars could move backward after reaching their preferred parking spot. Since a car was unable to park, $(2, 2, 2)$ is not a parking function.

It is well-known that the number of classical parking functions of length n is

$(n+1)^{n-1}$. This number surfaces in a variety of places; for example, it counts the number of labeled rooted forests on n vertices and the number of regions of a Shi arrangement (see [72] for further discussion).

Let PF_n denote the convex hull in $\mathbb{R}^n$ of all classical parking functions of length n; this is a polytope known as the *parking-function polytope*. In 2020, Stanley [62] asked for the volume and the number of vertices, faces, and lattice points of PF_n. These questions were first answered by Amanbayeva and Wang in [1].

Many of the questions asked about classical parking functions and their polytopes can also be asked about generalizations of parking functions. This chapter presents results on the combinatorial and geometric properties of the convex hull of a specific generalization of classical parking functions, namely **b**-parking functions.

Definition 8.1.3. Let $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{Z}^n_{>0}$. A **b-parking function** is a list $(\beta_1, \ldots, \beta_n)$ of positive integers whose non-decreasing rearrangement $\beta'_1 \leq \beta'_2 \leq \cdots \leq \beta'_n$ satisfies $\beta'_i \leq b_1 + \cdots + b_i$.

Classical parking functions are a special variety of **b**-parking functions for which the vector $\mathbf{b} = (b_1, \ldots, b_n)$ is the all-ones vector, i.e. $b_i = 1$ for all $1 \leq i \leq n$. In this special case, the partial sum which bounds the entries of the nondecreasing rearrangement of a **b**-parking function will always be $b_1 + \cdots + b_i = i$. This also means that for any integer n, a classical parking function of length n is also a **b**-parking function for any vector $\mathbf{b} \in \mathbb{Z}^n_{>0}$, since the partial sums for any such vector will be bounded below by the partial sums for the all-ones vector.

This generalization of parking functions has been previously explored from an enumerative perspective by Yan [70–72] and Pitman and Stanley [63]. In [36], Hanada, Lentfer, and Vindas-Meléndez generalized the classical parking-function polytope to the $\mathbf{b}$-*parking-function polytope* $\mathfrak{X}_n(\mathbf{b})$, i.e., the convex hull of all $\mathbf{b}$-parking functions of length n in $\mathbb{R}^n$ for $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{Z}^n_{>0}$. The work of Hanada, Lentfer, and Vindas-Meléndez focused on the $\mathbf{b}$-parking-function polytope in the special case when $\mathbf{b} = (a, b, b, \ldots, b)$. In particular, they described the face structure and provided a closed volume formula for the $\mathbf{b}$-parking-function polytope in that special case. These particular $\mathbf{b}$-parking-function polytopes have surprisingly arisen in different guises and in connection to other polytopes and combinatorial objects, including partial permutahedra [9,37], Pitman-Stanley polytopes [36,63], polytopes of win vectors [3,7], and stochastic sandpile models [60].

The results presented here explore $\mathbf{b}$-parking-function polytopes for general $\mathbf{b}$. Our contributions about $\mathbf{b}$-parking-function polytopes lend themselves to fruitful exploration from different perspectives and propose connections to seemingly unrelated combinatorial objects. We present definitions, terminology, and references in each relevant section. The chapter is structured as follows:

- We begin Section 8.2 by studying the $\mathbf{b}$-parking-function polytope from a polyhedral perspective. We show that $\mathfrak{X}_n(\mathbf{b})$ is (apart from a single degenerate case) an n-dimensional simple polytope (Proposition 8.2.2 and Theorem 8.2.3(a)), and we give explicit descriptions of the vertices, facets, and edges of $\mathfrak{X}_n(\mathbf{b})$ (Theorems 8.2.3(b) and 8.2.3(c) and Proposition 8.2.6, respectively). We then show that, up to a unimodular linear isomorphism, $\mathfrak{X}_n(\mathbf{b})$ is a generalized permutahedron (Theorem 8.2.11). We conclude the section by presenting the h-polynomial and f-vector of $\mathfrak{X}_n(\mathbf{b})$ (Theorem 8.2.15 and Corollary 8.2.16, respectively).

- In Section 8.3 we take a building-set perspective to study $\mathfrak{X}_n(\mathbf{b})$, which allows us to completely characterize the face structure of $\mathfrak{X}_n(\mathbf{b})$ for any $\mathbf{b} \in \mathbb{Z}^n_{>0}$ (Theorem 8.3.12). As a consequence, we show that, up to combinatorial equivalence, every $\mathbf{b}$-parking-function polytope is either the classical parking-function polytope PF_n or the stellohedron St_n (Corollary 8.3.14).

- We identify $\mathfrak{X}_n(\mathbf{b})$ as a special polymatroid (Theorem 8.4.3) in Section 8.4 and use this perspective to derive linear bounds on its combinatorial and circuit diameters (Theorem 8.4.4 and Theorem 8.4.6, respectively).

- In Section 8.5, we revisit the classical parking-function polytope PF_n and summarize its known and new connections to other established families of polytopes. We conclude by adding a description of PF_n as a projection of a relaxed Birkhoff polytope (Theorem 8.5.1), and $\mathfrak{X}_n(\mathbf{b})$ as a projection of a relaxed-partition polytope (Corollary 8.5.2).

8.2 Polyhedral perspectives on $\mathfrak{X}_n(\mathbf{b})$

Fix, throughout this work, a positive integer n and a vector $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{Z}^n_{>0}$. The following definition introduces our main object of study, i.e., a polytope associated to $\mathbf{b}$-parking functions, which we denote $\mathfrak{X}_n(\mathbf{b})$. Standard references on polytopes include [33] and [74].

Definition 8.2.1. The **b-parking-function polytope** $\mathfrak{X}_n(\mathbf{b})$ is the convex hull of all $\mathbf{b}$-parking functions in $\mathbb{R}^n$.

Recall that the linear action of the symmetric group $\mathfrak{S}_n$ on $\mathbb{R}^n$ is defined by $\pi\left(\sum_{i=1}^n a_i\, \mathbf{e}_i\right) := \sum_{i=1}^n a_i\, \mathbf{e}_{\pi(i)}$ for all $\pi \in \mathfrak{S}_n$, where $\mathbf{e}_1, \ldots, \mathbf{e}_n$ is the standard basis of $\mathbb{R}^n$. Note that by definition, the polytope $\mathfrak{X}_n(\mathbf{b})$ is symmetric under the action of $\mathfrak{S}_n$. For convenience, we use the notation

$$S_i := \sum_{j=1}^{i} b_j \text{ for } 1 \leq i \leq n \quad \text{and} \quad \mathbf{v}_k := (1, \ldots, 1, S_{k+1}, S_{k+2}, \ldots, S_n) \in \mathbb{R}^n \text{ for } 0 \leq k \leq n.$$

Further observe that the $\mathbf{v}_k$ are $\mathbf{b}$-parking functions, so $\mathbf{v}_k \in \mathfrak{X}_n(\mathbf{b})$ for all k. We will see below that the permutations $\pi(\mathbf{v}_k)$ of the $\mathbf{v}_k$ are precisely the vertices of $\mathfrak{X}_n(\mathbf{b})$ (Theorem 8.2.3(b)).

A theme that will appear early in our study of $\mathfrak{X}_n(\mathbf{b})$ is that the case in which $b_1 = 1$ behaves differently from when $b_1 \geq 2$. Indeed, observe that, if $b_1 \geq 2$, or if $b_1 = 1$ and $k \geq 2$, then the index k in $\mathbf{v}_k$ is precisely the number of entries in $\mathbf{v}_k$ that equal 1. In particular, when $b_1 \geq 2$, the $\mathbf{b}$-parking functions $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$ are distinct. However, if $b_1 = 1$, then $\mathbf{v}_0 = \mathbf{v}_1 = (1, S_2, S_3, \ldots, S_n)$. This trivial coincidence has important consequences for the structure of the polytope $\mathfrak{X}_n(\mathbf{b})$. Figure 8.1 shows the $\mathbf{b}$-parking-function polytopes $\mathfrak{X}_3(1,2,3)$ (left) and $\mathfrak{X}_3(2,3,4)$ (right). Note also that many lattice points in these polytopes are not parking functions. For example, $(2,2,2)$ is in $\mathfrak{X}_3(1,2,3)$, but is not a $\mathbf{b}$-parking function for $\mathbf{b} = (1,2,3)$.

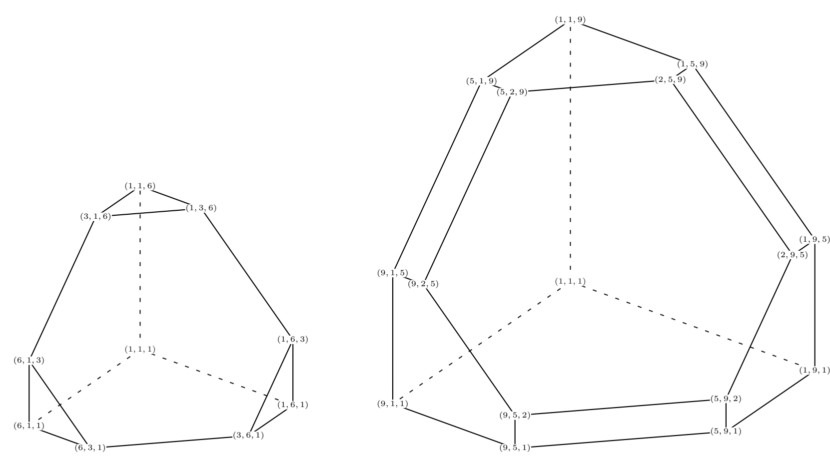


Figure 8.1: The polytopes $\mathfrak{X}_3(1,2,3)$ (left) and $\mathfrak{X}_3(2,3,4)$ (right).

We begin by proving that, with the exception of a single "degenerate" case, $\mathfrak{X}_n(\mathbf{b})$ is an n-dimensional polytope.

Proposition 8.2.2. *For all n, $\dim(\mathfrak{X}_n(\mathbf{b})) = n$, unless $n = 1 = b_1$, in which case $\dim(\mathfrak{X}_n(\mathbf{b})) = 0$.*

Proof. In the exceptional case $n = 1 = b_1$, the polytope $\mathfrak{X}_n(\mathbf{b})$ contains only the point (1) in $\mathbb{R}^1$. In all other cases, $\mathfrak{X}_n(\mathbf{b})$ is n-dimensional because, if $b_1 \geq 2$, then $\{\mathbf{v}_k : 0 \leq k \leq n\}$ is a set of $n+1$ affinely independent points in $\mathfrak{X}_n(\mathbf{b})$, and, if $b_1 = 1$ but $n \geq 2$, then $\{(S_2, 1, S_3, \ldots, S_n)\} \cup \{\mathbf{v}_k : 1 \leq k \leq n\}$ is a set of $n+1$ affinely independent points in $\mathfrak{X}_n(\mathbf{b})$. $\square$

In view of Proposition 8.2.2, we exclude the degenerate case $n = 1 = b_1$ from this point forward. That is, we proceed throughout the rest of this paper with the additional assumption that either $n \geq 2$ or, if $n = 1$, then $b_1 \geq 2$. Thus, $\dim \mathfrak{X}_n(\mathbf{b}) = n$.

8.2.1 Vertices, facets, and edges of $\mathfrak{X}_n(\mathbf{b})$

In this subsection, we describe the vertices, facets, and edges of the $\mathbf{b}$-parking-function polytope $\mathfrak{X}_n(\mathbf{b})$. In particular, we find that $\mathfrak{X}_n(\mathbf{b})$ is a simple polytope, meaning that the number of edges of $\mathfrak{X}_n(\mathbf{b})$ that are incident to each vertex of $\mathfrak{X}_n(\mathbf{b})$ is n.

Theorem 8.2.3.

(a) The polytope $\mathfrak{X}_n(\mathbf{b})$ is simple.

(b) The set of vertices of $\mathfrak{X}_n(\mathbf{b})$ *is* $\{\pi(\mathbf{v}_k) : \pi \in \mathfrak{S}_n$ *and* $0 \leq k \leq n\}$.

(c) The minimal *inequality description of* $\mathfrak{X}_n(\mathbf{b})$ *is the set of points* $(x_1, \ldots, x_n)$ *in* $\mathbb{R}^n$ *such that* $x_i \geq 1$ *for* $1 \leq i \leq n$ *and*

$$\sum_{i \in I} x_i \quad \leq \quad \sum_{i=n-|I|+1}^{n} S_i, \qquad (\mathbf{b}\text{-parking})$$

or, equivalently,

$$\sum_{i \in I} x_i \quad \leq \quad \sum_{j=1}^{n} \min\{j, |I|\}\, b_{n-j+1},$$

for all nonempty $I \subseteq [n]$ *if* $b_1 \geq 2$, *and for all nonempty* $I \subseteq [n]$ *with* $|I| \neq n-1$ *if* $b_1 = 1$. *In other words, these inequalities are precisely the facet-defining inequalities for* $\mathfrak{X}_n(\mathbf{b})$.

Before proceeding to the proof of Theorem 8.2.3, we note that, even when $b_1 = 1$, every point in $\mathfrak{X}_n(\mathbf{b})$ satisfies the ($\mathbf{b}$-parking) inequalities for *all* subsets $I \subseteq [n]$, including those where $|I| = n-1$. However, when $b_1 = 1$, these "$|I| = n-1$" ($\mathbf{b}$-parking) inequalities are redundant because they follow from the ($\mathbf{b}$-parking) inequality for $[n]$ together with the inequalities $x_i \geq 1$. Let $m \in [n]$ and $I := [n] \setminus \{m\}$. Then the ($\mathbf{b}$-parking) inequality for $[n]$ is

$$\sum_{i \in [n]} x_n \leq \sum_{j=1}^{n-1} j\, b_{n-j+1} + n.$$

Thus, subtracting the inequality $x_m \geq 1$ yields

$$\sum_{i \in I} x_i \leq \sum_{j=1}^{n-1} j\, b_{n-j+1} + (n-1) = \sum_{j=1}^{n} \min\{j, |I|\}\, b_{n-j+1},$$

which is the ($\mathbf{b}$-parking) inequality for I. Nonetheless, none of the other inequalities can be eliminated, as the following proof demonstrates *inter alia*.

We also remark here that Theorem 8.2.3 can be proved using the established theory of polymatroids. See Section 8.4 for further discussion of this connection. However, for completeness, we provide a proof here of Theorem 8.2.3 that does not rely on that theory. In fact, it is the details of the proof of Theorem 8.2.3 that motivate the forthcoming connections with generalized permutahedra and polymatroids.

Proof of Theorem 8.2.3. We temporarily write $P(\mathbf{b})$ for the polytope of points in $\mathbb{R}^n$ satisfying the system of inequalities given in Theorem 8.2.3(c). The outline of the proof is as follows. We first prove that $\mathfrak{X}_n(\mathbf{b}) \subseteq P(\mathbf{b})$. We then prove that the vertices of $P(\mathbf{b})$ are all of the form $\pi(\mathbf{v}_k)$ as in Theorem 8.2.3(b). Thus, the vertices of $P(\mathbf{b})$ are $\mathbf{b}$-parking functions, which proves the converse containment $P(\mathbf{b}) \subseteq \mathfrak{X}_n(\mathbf{b})$, giving us $\mathfrak{X}_n(\mathbf{b}) = P(\mathbf{b})$ and the "$\subseteq$" direction of Theorem 8.2.3(b). In the course of proving that every vertex of $P(\mathbf{b})$ is of the form $\pi(\mathbf{v}_k)$, we will have shown that each such vertex satisfies precisely n of the inequalities given in Theorem 8.2.3(c), which demonstrates that $\mathfrak{X}_n(\mathbf{b})$ is simple (Theorem 8.2.3(a)). We will then show that every point of the form $\pi(\mathbf{v}_k)$ is in fact a vertex of $\mathfrak{X}_n(\mathbf{b})$, completing the proof of Theorem 8.2.3(b). Finally, we will see that each of the inequalities in Theorem 8.2.3(c) is satisfied with equality by some point in $\mathfrak{X}_n(\mathbf{b})$. This fact, together with the fact that every vertex of $\mathfrak{X}_n(\mathbf{b})$ satisfies exactly n such inequalities, implies that the system of inequalities in Theorem 8.2.3(c) is minimal and thus will complete the proof of Theorem 8.2.3.

To prove that $\mathfrak{X}_n(\mathbf{b}) \subseteq P(\mathbf{b})$, it suffices to show that every vertex of $\mathfrak{X}_n(\mathbf{b})$ is in $P(\mathbf{b})$. To this end, let $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{R}^n$ be a vertex of $\mathfrak{X}_n(\mathbf{b})$, and let $I \subseteq [n]$. Write $\mathbf{v}' = (v'_1, \ldots, v'_n)$ for the weakly increasing reordering of $\mathbf{v}$. Thus, the final $|I|$ coordinates of $\mathbf{v}'$, namely $v'_{n-|I|+1}, v'_{n-|I|+2}, \ldots, v'_n$, are the $|I|$ weakly largest coordinates of $\mathbf{v}$. Moreover, $\mathbf{v}$ is a $\mathbf{b}$-parking function (because the vertices of the convex hull of a finite set are elements of that set), so $1 \le v'_i \le \sum_{j=1}^{i} b_j$ for $1 \le i \le n$. Hence, $v_i \ge 1$ for $1 \le i \le n$, and

$$\sum_{i \in I} v_i \;\le\; \sum_{i=n-|I|+1}^{n} v'_i \;\le\; \sum_{i=n-|I|+1}^{n} \sum_{j=1}^{i} b_j \;=\; \sum_{j=1}^{n} \sum_{i=\max\{j,\, n-|I|+1\}}^{n} b_j \;=\; \sum_{j=1}^{n} \min\{n-j+1, |I|\}\, b_j.$$

Reversing the order of summation yields that $\mathbf{v}$ satisfies Inequality ($\mathbf{b}$-parking). Thus, $\mathfrak{X}_n(\mathbf{b}) \subseteq P(\mathbf{b})$.

We now prove the converse containment. Let $\mathbf{w} = (w_1, \ldots, w_n)$ be a vertex of $P(\mathbf{b})$. It suffices to show that $\mathbf{w}$ is a $\mathbf{b}$-parking function. Since $P(\mathbf{b})$ and the set of $\mathbf{b}$-parking functions are both invariant under the action of $\mathfrak{S}_n$ on $\mathbb{R}^n$, we may suppose that $w_1 \le \cdots \le w_n$. We will show that $\mathbf{w} = \mathbf{v}_k$ for some k, making $\mathbf{w}$ evidently a $\mathbf{b}$-parking function.

Let $\mathcal{I}$ be the family of all nonempty subsets $I \subseteq [n]$ such that

$$\sum_{i \in I} w_i = \sum_{i=n-|I|+1}^{n} S_i, \tag{8.1}$$

and furthermore such that $|I| \ne n-1$ if $b_1 = 1$. Thus, $\mathcal{I}$ indexes the set of inequalities of the form ($\mathbf{b}$-parking) that the vertex $\mathbf{w}$ satisfies with equality. Let k be the number of coordinates of $\mathbf{w}$ that equal 1. Since $P(\mathbf{b})$ is a polytope in $\mathbb{R}^n$, and $\mathbf{w}$ is a vertex of this polytope, $\mathbf{w}$ satisfies with equality at least n of the inequalities that define $P(\mathbf{b})$. Hence, $|\mathcal{I}| \ge n-k$.

For each $I \subseteq [n]$, let $I_+ \coloneqq \{n-|I|+1, n-|I|+2, \ldots, n\}$, so that I_+ comprises the last $|I|$ elements of $[n]$. (We will see below that in fact $I = I_+$ for all $I \in \mathcal{I}$.)

Since

$$\sum_{i=n-|I|+1}^{n} S_i = \sum_{i \in I} w_i \le \sum_{i=n-|I|+1}^{n} w_i \le \sum_{i=n-|I|+1}^{n} S_i \qquad \text{for all } I \in \mathcal{I}$$

(where the middle inequality holds because $\mathbf{w}$ is weakly increasing, and the last inequality holds because $\mathbf{w} \in P(\mathbf{b})$), we have that

$$\sum_{i \in I} w_i = \sum_{i=n-|I|+1}^{n} w_i \qquad \text{for all } I \in \mathcal{I} \tag{8.2}$$

and, in particular, by Equation (8.1), that $I_+ \in \mathcal{I}$ for all $I \in \mathcal{I}$. Moreover, note that Equation (8.2) is an equation of two sums, each of which is a sum of the same number $|I|$ of terms, and that the ℓ^{th} term on the left-hand side of Equation (8.2) is bounded above by the ℓ^{th} term on the right-hand side for $1 \le \ell \le |I|$. It follows that the ℓ^{th} term on the left-hand side equals the ℓ^{th} term on the right-hand side for $1 \le \ell \le |I|$. In particular, the first terms on each side are equal:

$$w_{\min(I)} = w_{n-|I|+1} \qquad \text{for all } I \in \mathcal{I}. \tag{8.3}$$

Since $\mathbf{w} \in P(\mathbf{b})$, we have in particular that

$$\sum_{i=n-|I|+2}^{n} w_i \le \sum_{i=n-|I|+2}^{n} S_i \qquad \text{for all } I \subseteq [n], \tag{8.4}$$

while, from Equations (8.1) and (8.2),

$$\sum_{i=n-|I|+1}^{n} w_i = \sum_{i=n-|I|+1}^{n} S_i \qquad \text{for all } I \in \mathcal{I}. \tag{8.5}$$

Subtracting Inequality (8.4) from Equation (8.5) yields that

$$w_{n-|I|+1} \geq S_{n-|I|+1} \qquad \text{for all } I \in \mathcal{I}. \tag{8.6}$$

Moreover, since $\mathbf{w} \in P(\mathbf{b})$,

$$\sum_{i=n-|I|}^{n} w_i \leq \sum_{i=n-|I|}^{n} S_i \qquad \text{for all } I \subseteq [n] \text{ such that } |I| \leq n-1. \tag{8.7}$$

Subtracting Equation (8.5) from Inequality (8.7) yields

$$w_{n-|I|} \leq S_{n-|I|} \qquad \text{for all } I \in \mathcal{I} \text{ such that } |I| \leq n-1. \tag{8.8}$$

Since $S_{n-|I|} < S_{n-|I|+1}$, it follows from Inequalities (8.6) and (8.8) that

$$w_{n-|I|} < w_{n-|I|+1} \qquad \text{for all } I \in \mathcal{I} \text{ such that } |I| \leq n-1. \tag{8.9}$$

We now get that, in fact, $I = I_+$ for all $I \in \mathcal{I}$, since otherwise we would have an $I \in \mathcal{I}$ with $|I| \leq n-1$ and $\min(I) < \min(I_+) = n-|I|+1$, and so, from Equation (8.3) and $w_{\min(I)} \leq w_{n-|I|} \leq w_{n-|I|+1}$, we would get that $w_{n-|I|} = w_{n-|I|+1}$, contradicting Inequality (8.9). Thus, every element of $\mathcal{I}$ is of the form $I = I_j \coloneqq \{j, j+1, \ldots, n\}$ for some $1 \leq j \leq n$.

We will now show that $\mathbf{w} = \mathbf{v}_k = (1, \ldots, 1, S_{k+1}, S_{k+2}, \ldots, S_n)$. However, we must separately consider two cases. For the first case, suppose that $w_j > 1$ for all j such that $I_j \in \mathcal{I}$. Since $w_1 = \cdots = w_k = 1$, it follows that $j \geq k+1$ for all j such that $I_j \in \mathcal{I}$. In other words, every $I \in \mathcal{I}$ is of the form I_j with $k+1 \leq j \leq n$. But there are only $n-k$ such subsets of $[n]$, and $|\mathcal{I}| \geq n-k$, so we must have that $\mathcal{I}$ is *precisely* the set $\{I_{k+1}, I_{k+2}, \ldots, I_n\}$. Subtracting the (**b**-parking) *equation* for I_{j+1} from the equation for I_j, we get that $w_j = S_j$ for $k+1 \le j \le n-1$, and the I_n equation itself states that $w_n = S_n$. Thus, $\mathbf{w} = \mathbf{v}_k \in \mathfrak{X}_n(\mathbf{b})$, and moreover $\mathbf{w}$ satisfies precisely $k + |\mathcal{I}| = n$ of the inequalities in Theorem 8.2.3(c) with equality.

This leaves the case in which $w_j = 1$ for some j such that $I_j \in \mathcal{I}$. From Inequality (8.6), it follows that $S_j = 1$. Since $1 \le S_1 < \cdots < S_n$, we get that $j = 1$ and hence $b_1 = S_1 = 1 = w_1$ and $[n] = I_1 \in \mathcal{I}$. (Recall that we are also now assuming that $n \ge 2$ because we have excluded the degenerate case $n = 1 = b_1$ in the remarks following Proposition 8.2.2.) Furthermore, $k = 1$ because, since $[n] = I_1 \in \mathcal{I}$, we may subtract the ($\mathbf{b}$-parking) inequality for I_k from the ($\mathbf{b}$-parking) equation for $[n]$ to get that

$$k = \sum_{i=1}^{k} w_i \ge \sum_{i=1}^{k} S_i \ge \sum_{i=1}^{k} i,$$

which, together with $k \ge 1$ (because $w_1 = 1$), implies that $k = 1$. Thus, $|\mathcal{I}| \ge n-1$. Now, since $b_1 = 1$, the set $\mathcal{I}$ does not contain the subset I_2 because $|I_2| = n-1$, so $\mathcal{I}$ must contain all $n-2$ subsets $I_3, \ldots, I_n$.[1] Hence, $\mathcal{I} = \{[n], I_3, \ldots, I_n\}$. Subtracting the ($\mathbf{b}$-parking) equation for I_3 from the ($\mathbf{b}$-parking) equation for $[n]$ and applying $w_1 = 1 = b_1$ yields $w_2 = S_2$. For $3 \le j \le n-1$, subtract the ($\mathbf{b}$-parking) equation for I_{j+1} from the equation for I_j to get that $w_j = S_j$. Finally, the I_n equation itself states that $w_n = S_n$. Thus, we have again found that $\mathbf{w} = \mathbf{v}_k \in \mathfrak{X}_n(\mathbf{b})$ and that $\mathbf{w}$ satisfies precisely $k + |\mathcal{I}| = 1 + 1 + (n-2) = n$ of the inequalities in Theorem 8.2.3(c) with equality.

Since, in both cases, $\mathbf{w} \in \mathfrak{X}_n(\mathbf{b})$, we now have that $P(\mathbf{b}) = \mathfrak{X}_n(\mathbf{b})$. We have also found that every vertex of $P(\mathbf{b})$ equals $\pi(\mathbf{v}_k)$ for some $\pi \in \mathfrak{S}_n$ and $0 \le k \le n$, so we have now proved that every vertex of $\mathfrak{X}_n(\mathbf{b})$ has this same form. In addition, we have found that the number of inequalities in Theorem 8.2.3(c) that the vertex $\mathbf{w}$ satisfies

[1] The claim that there are exactly $n-2$ such subsets is the point at which this argument fails in the degenerate case $n = 1 = b_1$.

with equality is precisely n, so $\mathfrak{X}_n(\mathbf{b})$ is simple (proving Theorem 8.2.3(a)) and moreover no inequality given in Theorem 8.2.3(c) that supports $\mathfrak{X}_n(\mathbf{b})$ is redundant. In particular, since every point of the form $\mathbf{v}_k$ with $0 \leq k \leq n$ satisfies precisely n of the inequalities in Theorem 8.2.3(c) with equality, it follows that every such point is a vertex of $\mathfrak{X}_n(\mathbf{b})$. This completes the proof of Theorem 8.2.3(b).

The final claim is that this is a minimal system of inequalities. Since $\mathfrak{X}_n(\mathbf{b})$ is simple, it remains only to show that every inequality in this system supports $\mathfrak{X}_n(\mathbf{b})$. Of course, the inequalities of the form $x_i \geq 1$ are all satisfied with equality by the point $(1, \ldots, 1) \in \mathfrak{X}_n(\mathbf{b})$. For the ($\mathbf{b}$-parking) inequalities, let $I \subseteq [n]$ be given, and let $k \coloneqq n - |I|$. Then $\mathbf{v}_k$ satisfies the ($\mathbf{b}$-parking) inequality for I_+ with equality. Let $\pi \in \mathfrak{S}_n$ be any permutation under which the image of I_+ is I. Then $\pi(\mathbf{v}_k)$ is a point in $\mathfrak{X}_n(\mathbf{b})$ that satisfies the ($\mathbf{b}$-parking) inequality for I itself with equality. $\square$

As a corollary of Theorem 8.2.3(b), we enumerate the vertices of $\mathfrak{X}_n(\mathbf{b})$. When $b_1 = 1$ (and so $\mathbf{v}_0 = \mathbf{v}_1$), the number of vertices is the same as in the case of the classical parking-function polytope PF_n. However, when $b_1 \geq 2$, an additional $n!$ vertices come into play.

Corollary 8.2.4. *If $b_1 = 1$, then the number of vertices of $\mathfrak{X}_n(\mathbf{b})$ is $n!\left(\frac{1}{1!} + \cdots + \frac{1}{n!}\right)$. Otherwise, if $b_1 \geq 2$, then the number of vertices of $\mathfrak{X}_n(\mathbf{b})$ is $n!\left(\frac{1}{0!} + \frac{1}{1!} + \cdots + \frac{1}{n!}\right)$.*

Proof. When $b_1 \geq 2$, or when $b_1 = 1$ and $k \geq 2$, each vertex $\mathbf{v}$ of $\mathfrak{X}_n(\mathbf{b})$ of the form $\pi(\mathbf{v}_k)$ ($\pi \in \mathfrak{S}_n$) corresponds to the unique pair (I, σ) of the $(n-k)$-subset $I \subseteq [n]$ of entries of $\mathbf{v}$ that do not equal 1 and the bijection $\sigma \colon [n] \setminus [k] \to I$ that gives the position $\pi(j)$ of the value S_j in $\mathbf{v}$ for $k+1 \leq j \leq n$. On the other hand, when $b_1 = 1$ and $k \in \{0, 1\}$, so that $\mathbf{v}_0 = \mathbf{v}_1$, each vertex of the form $\pi(\mathbf{v}_k)$ corresponds to the unique permutation in $\mathfrak{S}_n$ (namely, π) that gives the position $\pi(j)$ of the value S_j in

$\mathbf{v}$ for $1 \le j \le n$. Thus, when $b_1 = 1$, the number of vertices is $n! + \sum_{k=2}^{n} \binom{n}{n-k}(n-k)!$, while, when $b_1 \ge 2$, the number of vertices is $\sum_{k=0}^{n} \binom{n}{n-k}(n-k)!$. □

From the characterization of the vertices and facets of $\mathfrak{X}_n(\mathbf{b})$, it is straightforward to describe the facets that contain a given nondecreasing vertex $\mathbf{v}_k$. To this end, define the following notation:

$$L_j \coloneqq \{(x_1, \dots, x_n) \in \mathfrak{X}_n(\mathbf{b}) \,:\, x_j = 1\} \qquad \text{for } 1 \le j \le n,$$
$$U_j \coloneqq \{(x_1, \dots, x_n) \in \mathfrak{X}_n(\mathbf{b}) \,:\, \textstyle\sum_{i=j}^{n} x_i = \sum_{i=j}^{n} S_i\} \qquad \text{for } 1 \le j \le n, \text{ with } j \ne 2 \text{ if } b_1 = 1.$$

Thus, L_j is the facet of $\mathfrak{X}_n(\mathbf{b})$ defined by the inequality $x_j \ge 1$, and U_j is the facet of $\mathfrak{X}_n(\mathbf{b})$ defined by the ($\mathbf{b}$-parking) inequality for the subset $I = I_j \coloneqq \{j, j+1, \dots, n\}$ (where $j \ne 2$ if $b_1 = 1$). In particular, it is routine to check that the facets that contain the vertex $\mathbf{v}_k$ are as follows.

Lemma 8.2.5.

(a) If $b_1 \ge 2$, or if $b_1 = 1$ and $k \ge 2$, then the n distinct facets that contain $\mathbf{v}_k$ are

$$L_1, L_2, \dots, L_k, U_{k+1}, U_{k+2}, \dots, U_n.$$

*(b) If $b_1 = 1$ and $k \in \{0, 1\}$ (*i.e.*, $\mathbf{v}_k = \mathbf{v}_0 = \mathbf{v}_1$), then the n distinct facets that contain $\mathbf{v}_k$ are*

$$L_1, U_1, U_3, U_4, \dots, U_n.$$

Proof. Since $\mathfrak{X}_n(\mathbf{b})$ is simple and n-dimensional, it suffices to confirm that the n given facets contain $\mathbf{v}_k$. □

We will now use Lemma 8.2.5 to describe the precise conditions under which two vertices of $\mathfrak{X}_n(\mathbf{b})$ share an edge. An informal summary of the conditions is as follows: Two vertices $\mathbf{v}$ and $\mathbf{w}$ share an edge if and only if either

1. the vertices are the same, except that the minimum entry > 1 in one vertex is a 1 in the other vertex; or
2. the vertices are the same, except that two entries of the form S_{j-1} and S_j swap positions.

In addition, we find that the edge vectors are all parallel to roots in the root system B_n, with lengths given directly by the entries of the vector $\mathbf{b}$ defining $\mathfrak{X}_n(\mathbf{b})$ or by their partial sums.

Proposition 8.2.6. *Let $\mathbf{v}$ and $\mathbf{w}$ be vertices of $\mathfrak{X}_n(\mathbf{b})$. Fix $\pi \in \mathfrak{S}_n$ and $k \in \mathbb{Z}$ with $0 \leq k \leq n$ such that $\mathbf{v} = \pi(\mathbf{v}_k)$.*

(a) If $b_1 \geq 2$, or if $b_1 = 1$ and $k \geq 2$, then $\mathbf{v}$ and $\mathbf{w}$ share an edge of $\mathfrak{X}_n(\mathbf{b})$ if and only if either

(1) $\mathbf{w} = (\pi \circ (j,k))(\mathbf{v}_{k-1})$ for some $1 \leq j \leq k$, in which case

$$\mathbf{w} - \mathbf{v} = (S_k - 1)\mathbf{e}_{\pi(j)}$$

(informally, the vertices are the same, except that one of the 1 entries in $\mathbf{v}$ is S_k in $\mathbf{w}$); or

(2) $\mathbf{w} = (\pi \circ (j-1,j))(\mathbf{v}_k)$ for some $k+2 \leq j \leq n$, in which case

$$\mathbf{w} - \mathbf{v} = b_j(\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)})$$

(informally, the vertices are the same, except that entries S_{j-1} and S_j swap positions);[2] *or*

(3) $k \leq n-1$ *and* $\mathbf{w} = \pi(\mathbf{v}_{k+1})$*, in which case*

$$\mathbf{w} - \mathbf{v} = -(S_{k+1} - 1)\mathbf{e}_{\pi(k+1)}$$

(informally, the vertices are the same, except that the S_{k+1} *entry in* $\mathbf{v}$ *is a* 1 *in* $\mathbf{w}$*).*[3]

(b) If $b_1 = 1$ *and* $k \in \{0, 1\}$ *(*i.e., $\mathbf{v}_k = \mathbf{v}_0 = \mathbf{v}_1$*), then* $\mathbf{v}$ *and* $\mathbf{w}$ *share an edge of* $\mathfrak{X}_n(\mathbf{b})$ *if and only if either*

(1) $\mathbf{w} = (\pi \circ (j-1, j))(\mathbf{v}_1)$ *for some* $2 \leq j \leq n$*, in which case*

$$\mathbf{w} - \mathbf{v} = b_j(\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)})$$

(informally, the vertices are the same, except that entries S_{j-1} *and* S_j *swap positions); or*

(2) $\mathbf{w} = \pi(\mathbf{v}_2)$*, in which case*

$$\mathbf{w} - \mathbf{v} = -b_2\mathbf{e}_{\pi(2)}$$

(informally, the vertices are the same, except that the S_2 *entry in* $\mathbf{v}$ *is a* 1 *in* $\mathbf{w}$*).*

Proof. For both parts of the proposition to be proved, note that we will be done once we have proved the "if" direction, since we will then have found n vertices that

[2] When $b_1 = 1$ and $k \in \{0, 1\}$ (so that $\mathbf{v}_k = \mathbf{v}_0 = \mathbf{v}_1$), we must use $k = 0$ for this equation to hold.

[3] When $b_1 = 1$ and $k \in \{0, 1\}$ (so that $\mathbf{v}_k = \mathbf{v}_0 = \mathbf{v}_1$), we must use $k = 1$ for this equation to hold.

are adjacent to $\mathbf{v}$, which must be all of the vertices that are adjacent to $\mathbf{v}$, because $\mathfrak{X}_n(\mathbf{b})$ is simple and of dimension n by Theorem 8.2.3(a) and Proposition 8.2.2. Thus, the "only if" direction will follow immediately. Note also that the calculation of the difference vector $\mathbf{w} - \mathbf{v}$ under each condition is straightforward.

To prove the "if" direction of Proposition 8.2.6(a), suppose that $b_1 \geq 2$, or $b_1 = 1$ and $k \geq 2$, as in Lemma 8.2.5(a). Thus, the facets that contain $\mathbf{v}_k$ are

$$L_1, L_2, \ldots, L_k, U_{k+1}, U_{k+2}, \ldots, U_n. \tag{8.10}$$

By the $\mathfrak{S}_n$ symmetry of $\mathfrak{X}_n(\mathbf{b})$, the vertices $\mathbf{v}$ and $\mathbf{w}$ share an edge if and only if $\pi^{-1}(\mathbf{w})$ lies on all but one of these facets. Now, if $\mathbf{w} = (\pi \circ (j,k))(\mathbf{v}_{k-1})$ for some $1 \leq j \leq k$, then $\pi^{-1}(\mathbf{w}) = (j,k)(\mathbf{v}_{k-1})$ lies on all facets in (8.10) except L_j. If $\mathbf{w} = (\pi \circ (j-1,j))(\mathbf{v}_k)$ for some $k+2 \leq j \leq n$, then $\pi^{-1}(\mathbf{w}) = (j-1,j)(\mathbf{v}_k)$ lies on all facets in (8.10) except U_j. Finally, if $k \leq n-1$ and $\mathbf{w} = \pi(\mathbf{v}_{k+1})$, then $\pi^{-1}(\mathbf{w}) = \mathbf{v}_{k+1}$ lies on all facets in (8.10) except U_{k+1}.

To prove the "if" direction of Proposition 8.2.6(b), suppose that $b_1 = 1$ and $k \in \{0,1\}$, as in Lemma 8.2.5(b). Thus, the facets that contain $\mathbf{v}_k = \mathbf{v}_0 = \mathbf{v}_1$ are

$$L_1, U_1, U_3, U_4, \ldots, U_n. \tag{8.11}$$

If $\mathbf{w} = (\pi \circ (1,2))(\mathbf{v}_0)$, then $\pi^{-1}(\mathbf{w}) = (1,2)(\mathbf{v}_0)$ lies on all facets in (8.11) except L_1. If $\mathbf{w} = (\pi \circ (j-1,j))(\mathbf{v}_0)$ for some $3 \leq j \leq n$, then $\pi^{-1}(\mathbf{w}) = (j-1,j)(\mathbf{v}_0)$ lies on all facets in (8.11) except U_j. Finally, if $\mathbf{w} = \pi(\mathbf{v}_2)$, then $\pi^{-1}(\mathbf{w}) = \mathbf{v}_2$ lies on all facets in (8.11) except U_1. $\square$

Recall that, given a vertex $\mathbf{v}$ of a polytope P, the *tangent cone* $\mathcal{T}_{\mathbf{v}}(P)$ at $\mathbf{v}$ is the cone generated by the edge vectors pointing from $\mathbf{v}$ to its neighbors in the edge graph of P. That is,

$$\mathcal{T}_{\mathbf{v}}(P) := \left\{ \sum_{\substack{\mathbf{w}\in \mathrm{Vert}(P):\\ \mathbf{w}\sim\mathbf{v}}} \lambda_{\mathbf{w}}(\mathbf{w}-\mathbf{v}) \;:\; \lambda_{\mathbf{w}} \geq 0 \text{ for all vertices } \mathbf{w}\sim\mathbf{v} \right\},$$

where we write $\mathbf{w} \sim \mathbf{v}$ when $\mathbf{w}$ and $\mathbf{v}$ are adjacent vertices of P. As a corollary of Proposition 8.2.6, we get an explicit description of the tangent cone at each vertex of $\mathfrak{X}_n(\mathbf{b})$.

Corollary 8.2.7. *Let $\mathbf{v}$ be a vertex of $\mathfrak{X}_n(\mathbf{b})$. Fix $\pi \in \mathfrak{S}_n$ and $k \in \mathbb{Z}$ with $0 \leq k \leq n$ such that $\mathbf{v} = \pi(\mathbf{v}_k)$. Let $\mathcal{T}_{\mathbf{v}} := \mathcal{T}_{\mathbf{v}}(\mathfrak{X}_n(\mathbf{b}))$ be the tangent cone at $\mathbf{v}$.*

(a) Suppose that $b_1 \geq 2$, or that $b_1 = 1$ and $k \geq 2$. Then $\mathcal{T}_{\mathbf{v}}$ is generated by

$$\left\{\mathbf{e}_{\pi(j)} \;:\; 1 \leq j \leq k\right\} \quad\cup\quad \left\{-\mathbf{e}_{\pi(k+1)}\right\} \quad\cup\quad \left\{\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)} \;:\; k+2 \leq j \leq n\right\}$$

if $k \leq n-1$, and by $\{\mathbf{e}_j \;:\; 1 \leq j \leq n\}$ if $k = n$.

(b) Suppose that $b_1 = 1$ and $k \in \{0, 1\}$. Then $\mathcal{T}_{\mathbf{v}}$ is generated by

$$\left\{\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)} \;:\; 2 \leq j \leq n\right\} \quad\cup\quad \left\{-\mathbf{e}_{\pi(2)}\right\}.$$

8.2.2 $\mathfrak{X}_n(\mathbf{b})$ is a generalized permutahedron

Generalized permutahedra (sometimes spelled *permutohedra*) form a widely studied class of polytopes with a rich theory and many interesting subclasses. One definition of a generalized permutahedron is that it is a polyhedron whose normal fan is a coarsening of the braid arrangement. For sources on generalized permutahedra, we recommend [26, 40, 55, 56]. We use an equivalent characterization.

Definition 8.2.8. A polytope is a **generalized permutahedron** if and only if all edges are parallel to $\mathbf{e}_i - \mathbf{e}_j$ for some standard basis vectors $\mathbf{e}_i$ and $\mathbf{e}_j$.

We show that the lifting of $\mathfrak{X}_n(\mathbf{b})$ is a generalized permutahedron, from which it, and $\mathfrak{X}_n(\mathbf{b})$ itself, immediately inherit a variety of combinatorial and geometric results and properties. Let $B \coloneqq \sum_{k=1}^{n} S_k$, and let $\ell \colon \mathbb{R}^n \to \mathbb{R}^{n+1}$ be the affine-linear map that "vertically projects" $\mathbb{R}^n$ onto the hyperplane $H = \{\mathbf{x} \in \mathbb{R}^{n+1} : x_1 + \cdots + x_n + x_{n+1} = B\}$ via $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, B - \sum_{i=1}^{n} x_i)$. Using this mapping, Proposition 8.2.10 below follows.

Definition 8.2.9. Let $\overline{\mathfrak{X}}_n(\mathbf{b}) \coloneqq \{\ell(\mathbf{x}) : \mathbf{x} \in \mathfrak{X}_n(\mathbf{b})\}$ be the lifting of $\mathfrak{X}_n(\mathbf{b})$ to the hyperplane H. We call $\overline{\mathfrak{X}}_n(\mathbf{b})$ the **lifted b-parking-function polytope**.

Proposition 8.2.10. *The vertex set of* $\overline{\mathfrak{X}}_n(\mathbf{b})$ *is* $\{\ell(\mathbf{v}) : \mathbf{v}$ *is a vertex of* $\mathfrak{X}_n(\mathbf{b})\}$. *Furthermore, the minimal inequality description of* $\overline{\mathfrak{X}}_n(\mathbf{b})$ *is the minimal inequality description of* $\mathfrak{X}_n(\mathbf{b})$ *together with the additional equality* $x_1 + \cdots + x_n + x_{n+1} = B$.

Building off this combinatorial structure, we have the main result of this subsection.

Theorem 8.2.11. *The lifted* $\mathbf{b}$*-parking-function polytope* $\overline{\mathfrak{X}}_n(\mathbf{b})$ *is a generalized permutahedron.*

Proof. Proposition 8.2.10 implies that the edges of $\mathfrak{X}_n(\mathbf{b})$ lift to edges of $\overline{\mathfrak{X}}_n(\mathbf{b})$ in the natural way: $\{\mathbf{x}, \mathbf{y}\}$ is an edge of $\mathfrak{X}_n(\mathbf{b})$ if and only if $\{\ell(\mathbf{x}), \ell(\mathbf{y})\}$ is an edge of $\overline{\mathfrak{X}}_n(\mathbf{b})$. If $\{\mathbf{x}, \mathbf{y}\}$ is an edge of $\mathfrak{X}_n(\mathbf{b})$ with $\mathbf{x} - \mathbf{y} = c(\mathbf{e}_i - \mathbf{e}_j)$, then $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$. Thus $\ell(\mathbf{x})_{n+1} = B - \sum_{i=1}^{n} x_i = B - \sum_{i=1}^{n} y_i$. so $\ell(\mathbf{x}) - \ell(\mathbf{y}) = c(\mathbf{e}_i - \mathbf{e}_j)$. If $\{\mathbf{x}, \mathbf{y}\}$ is an edge of $\mathfrak{X}_n(\mathbf{b})$ with $\mathbf{x} - \mathbf{y} = c\mathbf{e}_j$, then $\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i = c$. Thus, $\ell(\mathbf{x})_{n+1} = B - \sum_{i=1}^{n} x_i = B - \sum_{i=1}^{n} y_i - c = \ell(\mathbf{y})_{n+1} - c$, so $\ell(\mathbf{x}) - \ell(\mathbf{y}) = c(\mathbf{e}_j - \mathbf{e}_{n+1})$.

By Proposition 8.2.6, all edges of $\mathfrak{X}_n(\mathbf{b})$ are of this form. Hence, $\overline{\mathfrak{X}}_n(\mathbf{b})$ is a generalized permutahedron. □

We may now apply established results about generalized permutahedra to $\overline{\mathfrak{X}}_n(\mathbf{b})$. One such result is [40, Corollary 4.8], which provides a formula for the number of lattice points of a generalized permutahedron. Since lifting a polytope does not change the number of lattice points of a polytope, by applying the formula we obtain the number of lattice points of $\mathfrak{X}_n(\mathbf{b})$. Furthermore, we note that the enumeration of $\mathbf{b}$-parking functions is not completely known [70, 71], but by applying the formula of [40, Corollary 4.8], we obtain an upper-bound for the number of $\mathbf{b}$-parking functions since all $\mathbf{b}$-parking functions will be lattice points in $\mathfrak{X}_n(\mathbf{b})$.

Additionally, by a theorem of Backman and Liu [4], Theorem 8.2.11 implies that $\overline{\mathfrak{X}}_n(\mathbf{b})$, and hence $\mathfrak{X}_n(\mathbf{b})$ itself, admits a regular unimodular triangulation. For a definition of and further details about regular unimodular triangulations, see [35]. We can also express $\overline{\mathfrak{X}}_n(\mathbf{b})$ as a Minkowski sum of simplices. In order to do that, we begin by expressing $\overline{\mathfrak{X}}_n(\mathbf{b})$ in the notation introduced by Postnikov [56] for generalized permutahedra. Following from [56, Section 6], every generalized permutahedron can be expressed as

$$P_n^{\mathcal{Z}}(\{z_I\}) := \left\{ (x_1, \ldots, x_n) \in \mathbb{R}^n \,:\, \sum_{i=1}^{n} x_i = z_{[n]},\ \sum_{i\in I} x_i \geq z_I \ \text{ for any subset } I \subset [n] \right\},$$

for some collection of real parameters $\{z_I\}$ over subsets I of $[n]$ with $z_\varnothing = 0$.

Proposition 8.2.12. *The lifted* $\mathbf{b}$*-parking-function polytope* $\overline{\mathfrak{X}}_n(\mathbf{b})$ *is the generalized permutahedron* $P_{n+1}^{\mathcal{Z}}(\{z_I\})$ *in which the values of the parameters* z_I *for* $\varnothing \neq I \subseteq [n+1]$

are given by

$$z_I \coloneqq \begin{cases} |I| & \textit{if } n+1 \notin I, \\ S_1 + S_2 + \cdots + S_{|I|-1} & \textit{if } n+1 \in I. \end{cases}$$

Proof. From Proposition 8.2.10, consider the defining inequalities,

$$\sum_{i \in I} x_i \leq S_n + S_{n-1} + \cdots + S_{n-|I|+1},$$

for any I with $|I| = k$, $n+1 \notin I$, and $x_1 + \ldots + x_{n+1} = \sum_{i=1}^{n} S_i$. Subtracting these two yields the inequality

$$\sum_{i \in [n+1] \setminus I} x_i \geq S_1 + S_2 + \cdots + S_{|[n+1] \setminus I|-1}.$$

This shows that $\overline{\mathfrak{X}}_n(\mathbf{b})$ can be described in terms of the following inequalities:

$$\begin{cases} x_i \geq 1, & \text{for } 1 \leq i \leq n, \\ \displaystyle\sum_{i \in I} x_i \geq \sum_{i=1}^{|I|-1} S_i, & \text{for all } I \subset [n+1] \text{ with } n+1 \in I, \\ \displaystyle\sum_{i=1}^{n+1} x_i = \sum_{i=1}^{n} S_i, & \end{cases}$$

since they recover the defining inequalities in Theorem 8.2.3. To match with Postnikov's definition of $P_{n+1}^{\mathcal{Z}}(\{z_I\})$, we can add the following redundant inequalities obtained from $x_i \geq 1$ which have no effect on the solutions of the above system of inequalities:

$$\sum_{i \in I} x_i \geq |I| \quad \text{for all } I \subset [n+1] \text{ s.t. } n+1 \notin I.$$

These inequalities give all the values of z_I for all nonempty $I \subset [n+1]$. $\square$

In [2], Ardila, Benedetti, and Doker prove that any generalized permutahedron can be written as a signed Minkowski sum of scaled standard simplices. Specifically,

$$P_n^{\mathcal{Z}}(\{z_I\}) = P_n^{\mathcal{Y}}(\{y_I\}) \coloneqq \sum_{I \subseteq [n]} y_I \triangle_I,$$

where $y_I = \sum_{J \subseteq I} (-1)^{|I|-|J|} z_J$ and $\triangle_I \coloneqq \operatorname{conv}\{\mathbf{e}_i : i \in I\}$, for each $I \subseteq [n]$. In the case that $y_I \geq 0$ for all I, we call this a *$\mathcal{Y}$-generalized permutahedron*, which enjoys a variety of special properties including Ehrhart positivity and h^*-real-rootedness [56]; for more on these properties, see [17, 18, 50].

Proposition 8.2.13. *The lifted* $\mathbf{b}$*-parking-function polytope* $\overline{\mathfrak{X}}_n(\mathbf{b})$ *is the signed Minkowski sum* $\sum_{I \subseteq [n+1]} y_I \triangle_I$*, where*

$$y_I \coloneqq \begin{cases} 1 & \text{if } n+1 \notin I \text{ and } |I| = 1, \\ 0 & \text{if } I = \{n+1\}, \text{ or if } n+1 \notin I \text{ and } |I| \geq 2, \\ b_1 - 1 & \text{if } n+1 \in I \text{ and } |I| = 2, \\ \sum_{j=0}^{|I|-3} (-1)^{|I|+j-1} \binom{|I|-3}{j} b_{j+2} & \text{if } n+1 \in I,\ |I| \geq 3. \end{cases}$$

Proof. We write $\overline{\mathfrak{X}}_n(\mathbf{b}) = P_{n+1}^{\mathcal{Z}}(\{z_I\})$, with z_I as given in Proposition 8.2.12. By [26, Prop. 2.2.4], this becomes $P_{n+1}^{\mathcal{Z}}(\{z_I\}) = \sum_{I \subseteq [n+1]} y_I \triangle_I$, where the y_Is are determined by $z_I = \sum_{J \subseteq I} y_J$ for all nonempty $I \subseteq [n+1]$.

First note that $y_{\{i\}} = z_{\{i\}} = \begin{cases} 1 & \text{if } 1 \leq i \leq n, \\ 0 & \text{if } i = n+1. \end{cases}$

For $I \subseteq [n+1]$ not containing $n+1$ with $|I| \geq 2$, we apply the principle of inclusion–exclusion to get

$$y_I = \sum_{J \subseteq I} (-1)^{|I|-|J|} z_J = \sum_{k=0}^{|I|} (-1)^{|I|-k} \binom{|I|}{k} k = 0.$$

For $I \subseteq [n+1]$ with $n+1 \in I$, $|I| \geq 2$, then again by the principle of inclusion–exclusion,

$$\begin{aligned} y_I = \sum_{J \subseteq I} (-1)^{|I|-|J|} z_J &= \sum_{J \subseteq I \setminus \{n+1\}} (-1)^{|I|-|J|} z_J + \sum_{\substack{J \subset I \\ n+1 \in J}} (-1)^{|I|-|J|} z_J \\ &= -y_{I \setminus \{n+1\}} + \sum_{k=1}^{|I|-1} (-1)^{|I|-k-1} \binom{|I|-1}{k} \sum_{\ell=1}^{k} S_\ell. \end{aligned} \tag{8.12}$$

When $|I| = 2$, $y_I = -1 + b_1$. For $|I| \geq 3$, the second term on the right-hand side of Equation (8.12) can be rewritten as

$$\begin{aligned} \sum_{\ell=1}^{|I|-1} S_\ell \left(\sum_{k=\ell}^{|I|-1} (-1)^{|I|-1-k} \binom{|I|-1}{k} \right) &= \sum_{\ell=1}^{|I|-1} S_\ell \left(\sum_{k=0}^{|I|-1-\ell} (-1)^k \binom{|I|-1}{k} \right) \\ = \sum_{\ell=1}^{|I|-1} \left(\sum_{i=1}^{\ell} b_i \right) (-1)^{|I|-1-\ell} \binom{|I|-2}{|I|-1-\ell} &= \sum_{i=1}^{|I|-1} b_i \sum_{\ell=i}^{|I|-1} (-1)^{|I|-1-\ell} \binom{|I|-2}{|I|-1-\ell} \\ = \sum_{i=1}^{|I|-1} b_i \sum_{\ell=0}^{|I|-1-i} (-1)^{\ell} \binom{|I|-2}{\ell} &= \sum_{i=2}^{|I|-1} (-1)^{|I|-1-i} \binom{|I|-3}{|I|-1-i} b_i. \end{aligned}$$

Hence, for $|I| \geq 3$,

$$y_I = \sum_{i=2}^{|I|-1} (-1)^{|I|-1-i} \binom{|I|-3}{|I|-1-i} b_i = (-1)^{|I|-1} \sum_{i=0}^{|I|-3} (-1)^i \binom{|I|-3}{i} b_{i+2}. \qquad \square$$

Proposition 8.2.13 allows us to determine from $\mathbf{b} = (b_1, \ldots, b_n)$ whether $\overline{\mathfrak{X}}_n(\mathbf{b})$ is a $\mathcal{Y}$-generalized permutahedron. In particular, the liftings of PF_n and of the specific $\mathbf{b}$-parking-function polytope $\mathfrak{X}_n(a, b)$ studied in [36] (where $(a, b) \coloneqq \mathbf{b} = (a, b, b, \ldots, b)$) are both $\mathcal{Y}$-generalized permutahedra.

8.2.3 h-Polynomial of $\mathfrak{X}_n(\mathbf{b})$

We now consider the h-polynomial of the $\mathbf{b}$-parking-function polytope. Given an n-dimensional polytope P and $0 \leq i \leq n$, let $f_i(P)$ denote the number of i-dimensional faces of P. The f-polynomial of P is then defined as $f(P;t) \coloneqq \sum_{i=0}^{n} f_i(P)t^i$ and the h-polynomial of P is the result of applying a change of basis given by $h(P;t) \coloneqq f(P;t-1)$.

The stellohedron St_n is a simple generalized permutahedron first defined by Postnikov, Reiner, and Williams [55] as the nestohedron for a certain building set (see Section 8.3 for more details), which is intimately connected to $\mathfrak{X}_n(\mathbf{b})$. The h-polynomial of St_n is the *binomial Eulerian polynomial*

$$\widetilde{A}_n(z) = 1 + z\sum_{k=1}^{n}\binom{n}{k}A_k(z),$$

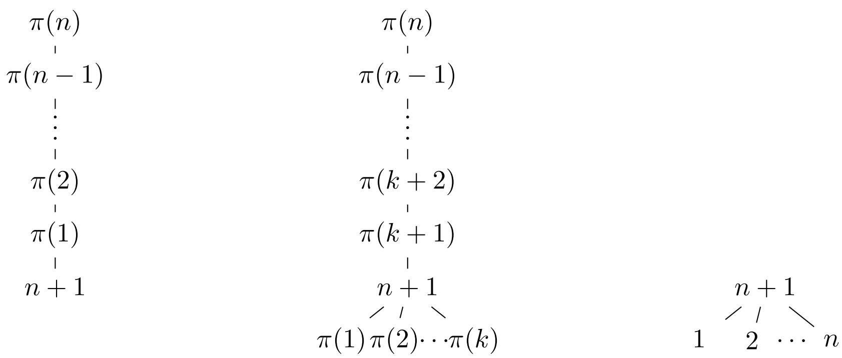
where $A_k(z) = \sum_{\sigma\in\mathfrak{S}_k} z^{\mathsf{des}(\sigma)}$ is the Eulerian polynomial [55].

From Theorem 8.2.3(a) and Theorem 8.2.11, we have that $\overline{\mathfrak{X}}_n(\mathbf{b})$ is a simple generalized permutahedron. By [55, Theorem 4.2], the h-polynomial of a simple generalized permutahedron can be computed using its so-called *vertex posets*, which are defined for every vertex of the polytope as follows.

For a generalized permutahedron P of dimension n in $\mathbb{R}^{n+1}$, the normal cone of every vertex v of P is the intersection of the half spaces in $\mathbb{R}^{n+1}/(1,\ldots,1)\mathbb{R}$ defined by a set of inequalities of the form $x_i \leq x_j$, where $(i,j) \in [n+1]^2$. One can associate to each vertex v a *vertex poset* Q_v by setting $i \leq_{Q_v} j$ whenever $x_i \leq x_j$ in the normal cone of v. We denote covering relations by $\lessdot$. The *descent set* $\mathsf{Des}(Q_v)$ of Q_v is the set of ordered pairs (i,j) such that $i \lessdot_{Q_v} j$ but $i >_{\mathbb{Z}} j$. The *number of descents* of Q_v is $\mathsf{des}(Q_v) := |\mathsf{Des}(Q_v)|$. When P is a simple generalized permutahedron, by [55, Theorem 4.2], its h-polynomial is given by

$$h(P;z) = \sum_{v \in \mathrm{Vert}(P)} z^{\mathsf{des}(Q_v)}.$$

Since the face numbers of $\mathfrak{X}_n(\mathbf{b})$ do not change after the lifting ℓ, we may use this equation to compute the h-polynomial of $\mathfrak{X}_n(\mathbf{b})$. We characterize the vertex poset for every vertex of $\overline{\mathfrak{X}}_n(\mathbf{b})$.



(a) The $k=0$ case (b) The $1 \le k \le n-1$ case (c) The $k=n$ case

Figure 8.2: The vertex poset $Q_{\overline{\mathbf{v}}}$ of $\overline{\mathfrak{X}}_n(\mathbf{b})$ in Lemma 8.2.14(a)

Lemma 8.2.14. *Let $\overline{\mathbf{v}}$ be a vertex of $\overline{\mathfrak{X}}_n(\mathbf{b})$. Fix $\pi \in \mathfrak{S}_n$ and $k \in \mathbb{Z}$ with $0 \le k \le n$ such that $\overline{\mathbf{v}}$ is the image under the lifting ℓ of the vertex $\pi(\mathbf{v}_k)$ of $\mathfrak{X}_n(\mathbf{b})$.*

(a) If $b_1 \ge 2$, or if $b_1 = 1$ and $k \ge 2$, then the vertex poset $Q_{\overline{\mathbf{v}}}$ is defined by the covering relations

$$\begin{aligned} &\pi(j) \lessdot n+1 && \text{for } 1 \le j \le k, \\ &n+1 \lessdot \pi(k+1), && \\ &\pi(j-1) \lessdot \pi(j) && \text{for } k+2 \le j \le n. \end{aligned}$$

(See the Hasse diagrams in Figure 8.2.)

(b) *If $b_1 = 1$ and $k \in \{0, 1\}$ (in which case $\mathbf{v}_k = \mathbf{v}_0 = \mathbf{v}_1$), then the vertex poset $Q_{\overline{\mathbf{v}}}$ is defined by the covering relations*

$$\begin{aligned} \pi(1) \lessdot \pi(2) \lessdot \cdots \lessdot \pi(n), \\ n+1 \lessdot \pi(2). \end{aligned}$$

(See the Hasse diagram in Figure 8.3.)

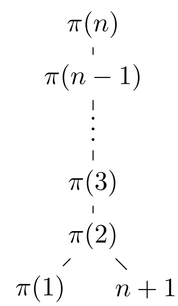


Figure 8.3: The vertex poset $Q_{\overline{\mathbf{v}}}$ of $\overline{\mathfrak{X}}_n(\mathbf{b})$ in Lemma 8.2.14(b)

Proof. Let $\mathbf{v} \coloneqq \pi(\mathbf{v}_k)$, so that $\overline{\mathbf{v}} = \ell(\mathbf{v})$. The normal cone at $\overline{\mathbf{v}}$ is the polar of the tangent cone at $\overline{\mathbf{v}}$. From the formula for the affine projection ℓ, we have that $\ell(\mathbf{w}) - \ell(\mathbf{v}) = L(\mathbf{w} - \mathbf{v})$ for all $\mathbf{w} \in \mathbb{R}^n$, where $L\colon \mathbb{R}^n \to \mathbb{R}^{n+1}$ is the linear map $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, -\sum_{i=1}^n x_i)$. To compute the tangent cone of $\overline{\mathfrak{X}}_n(\mathbf{b})$ at $\overline{\mathbf{v}}$, we apply L to the generators of $\mathcal{T}_{\mathbf{v}}(\mathfrak{X}_n(\mathbf{b}))$ given in Corollary 8.2.7. Now,

$$\begin{aligned} L(\mathbf{e}_{\pi(j)}) &= \mathbf{e}_{\pi(j)} - \mathbf{e}_{n+1}, \\ L(-\mathbf{e}_{\pi(j)}) &= -\mathbf{e}_{\pi(j)} + \mathbf{e}_{n+1}, \\ L(\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)}) &= \mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)}, \end{aligned}$$

for $1 \leq j \leq n$.[4] Thus, the tangent cone $\mathcal{T}_{\overline{\mathbf{v}}}$ of $\overline{\mathfrak{X}}_n(\mathbf{b})$ at $\overline{\mathbf{v}}$ is generated as follows:

[4]Note that, in an abuse of notation, we are writing $\mathbf{e}_j$ for the j^{th} standard basis vector in $\mathbb{R}^{n+1}$ on the right-hand side and in $\mathbb{R}^n$ on the left-hand side of the equations.

(a) If $b_1 \geq 2$, or if $b_1 = 1$ and $k \geq 2$, then $\mathcal{T}_{\overline{\mathbf{v}}}$ is generated by

$$\left\{\mathbf{e}_{\pi(j)} - \mathbf{e}_{n+1} \,:\, 1 \leq j \leq k\right\} \cup \left\{-\mathbf{e}_{\pi(k+1)} + \mathbf{e}_{n+1}\right\} \cup \left\{\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)} \,:\, k+2 \leq j \leq n\right\}$$

if $k \leq n-1$, and by $\{\mathbf{e}_j - \mathbf{e}_{n+1} \,:\, 1 \leq j \leq n\}$ if $k = n$.

(b) If $b_1 = 1$ and $k \in \{0, 1\}$, then $\mathcal{T}_{\overline{\mathbf{v}}}$ is generated by

$$\left\{\mathbf{e}_{\pi(j-1)} - \mathbf{e}_{\pi(j)} \,:\, 2 \leq j \leq n\right\} \cup \left\{-\mathbf{e}_{\pi(2)} + \mathbf{e}_{n+1}\right\}.$$

These generating vectors of the tangent cone are the facet-defining outer normals of the normal cone. Thus, for example, in the case where $b_1 \geq 2$ and $k \leq n-1$, the normal cone at $\overline{\mathbf{v}}$ is defined by the inequalities

$$\begin{aligned} x_{\pi(j)} &\leq x_{n+1} && \text{for } 1 \leq j \leq k, \\ x_{n+1} &\leq x_{\pi(k+1)}, && \\ x_{\pi(j-1)} &\leq x_{\pi(j)} && \text{for } k+2 \leq j \leq n; \end{aligned}$$

and the other cases are handled similarly. These inequalities determine the covering relations given in the statement of Lemma 8.2.14. □

Theorem 8.2.15. *The h-polynomial of $\mathfrak{X}_n(\mathbf{b})$ is*

$$h(\mathfrak{X}_n(\mathbf{b}); z) = \begin{cases} \widetilde{A}_n(z) - nzA_{n-1}(z) & \text{if } b_1 = 1, \\ \widetilde{A}_n(z) & \text{if } b_1 \geq 2. \end{cases}$$

Proof. Consider first the case in which $b_1 \geq 2$. From Figure 8.2 (corresponding to Lemma 8.2.14(a)), we see that $Q_{\overline{\pi(\mathbf{v}_k)}}$ contains $1 + \mathsf{des}(\pi(k+1), \ldots, \pi(n))$ descents if $0 \leq k \leq n-1$, and 0 descents if $k = n$. Thus, enumerating vertices as in the proof of

Corollary 8.2.4,

$$\sum_{\mathbf{v}\in \mathrm{Vert}(\mathfrak{X}_n(\mathbf{b}))} z^{\mathsf{des}(Q_{\overline{\mathbf{v}}})} = \sum_{k=0}^{n-1} \sum_{\substack{I\subseteq[n]:\\ |I|=n-k}} \sum_{\sigma\in\mathfrak{S}_{n-k}} z^{1+\mathsf{des}(\sigma)} + 1 = 1 + z\sum_{k=1}^{n} \binom{n}{k} \sum_{\sigma\in\mathfrak{S}_k} z^{\mathsf{des}(\sigma)} = \widetilde{A}_n(z).$$

Now consider the case in which $b_1 = 1$. From Figures 8.2 and 8.3, we see that $Q_{\overline{\pi(\mathbf{v}_k)}}$ contains $1 + \mathsf{des}(\pi)$ descents if $k = 1$, $1 + \mathsf{des}(\pi(k+1), \ldots, \pi(n))$ descents if $2 \le k \le n-1$, and 0 descents if $k = n$. Again enumerating vertices as in the proof of Corollary 8.2.4,

$$\begin{aligned}\sum_{\mathbf{v}\in \mathrm{Vert}(\mathfrak{X}_n(\mathbf{b}))} z^{\mathsf{des}(Q_{\overline{\mathbf{v}}})} &= \sum_{\pi\in\mathfrak{S}_n} z^{1+\mathsf{des}(\pi)} + \sum_{k=2}^{n-1} \sum_{\substack{I\subseteq[n]:\\ |I|=n-k}} \sum_{\sigma\in\mathfrak{S}_{n-k}} z^{1+\mathsf{des}(\sigma)} + 1 \\ &= 1 + z\sum_{k=1}^{n-2} \binom{n}{k} \sum_{\sigma\in\mathfrak{S}_k} z^{\mathsf{des}(\sigma)} + z\sum_{\pi\in\mathfrak{S}_n} z^{\mathsf{des}(\pi)} = \widetilde{A}_n(z) - nzA_{n-1}(z)\square\end{aligned}$$

Since the h-polynomial of $\mathfrak{X}_n(\mathbf{b})$ depends only on whether $b_1 = 1$ or $b_1 \ge 2$, comparing with results on faces numbers of $\mathfrak{X}_n(a, b, b, \ldots, b)$ in [36, Prop. 3.14], we immediately have the results on face numbers of $\mathfrak{X}_n(\mathbf{b})$ for all $\mathbf{b}$.

Corollary 8.2.16. *Let f_k be the number of k-dimensional faces of $\mathfrak{X}_n(\mathbf{b})$ for $k = 0, 1, \ldots, n$. Then,*

$$f_k = \begin{cases} \sum_{j=0, j\neq 1}^{n-k} \binom{n}{j}(n-k-j)!\, S(n-j+1, n-k-j+1) & \text{if } b_1 = 1, \\ \sum_{j=0}^{n-k} \binom{n}{j}(n-k-j)!\, S(n-j+1, n-k-j+1) & \text{if } b_1 \ge 2, \end{cases}$$

where $S(n,k)$ is the Stirling number of the second kind.

8.3 A building set perspective

In this section, we completely characterize the face structure and identify the combinatorial types of all **b**-parking-function polytopes. In particular, we show that even though not every **b**-parking-function polytope is a nestohedron (see Definition 8.3.2), up to combinatorial equivalence, every **b**-parking-function polytope is either the classical parking-function polytope PF_n (if $b_1 = 1$) or the stellohedron St_n (if $b_1 \geq 2$).

We consider the $\mathcal{Y}$-generalized permutahedron associated with a certain class of collections $\mathcal{B}$. We recall some background on building sets, nestohedra, and nested set complexes from [55, 56].

Definition 8.3.1 ([56, Definition 7.1])**.** Let E be a finite set. A **building set** on E is a collection $\mathcal{B}$ of nonempty subsets of E that satisfies the conditions:

(B1) If $I, J \in \mathcal{B}$ and $I \cap J \neq \varnothing$, then $I \cup J \in \mathcal{B}$.

(B2) The singleton $\{i\}$ lies in $\mathcal{B}$ for all $i \in E$.

In this case, we write $\mathcal{B}_{\max} \subset \mathcal{B}$ for the collection of maximal subsets in $\mathcal{B}$ under inclusion.

Definition 8.3.2. Let $\mathcal{B}$ be a building set in $[n]$. A **nestohedron** $P_{\mathcal{B}}$ is a $\mathcal{Y}$-generalized permutahedron with positive coefficients y_I only for $I \in \mathcal{B}$: $P_n^{\mathcal{Y}}(\{y_I\}) = \sum_{I \in \mathcal{B}} y_I \triangle_I$.

We will relate nestohedra to certain complexes formed from building sets.

Definition 8.3.3 ([55, Definition 6.4])**.** Let $\mathcal{B}$ be a building set on a finite set E. A **nested set** of $\mathcal{B}$ is a subset $N \subseteq \mathcal{B} \setminus \mathcal{B}_{\max}$ that satisfies the following conditions:

(N1) If $I, J \in N$, then either $I \subseteq J$ or $I \supseteq J$ or $I \cap J = \varnothing$.

(N2) If $J_1, \ldots, J_k \in N$ are pairwise disjoint with $k \geq 2$, then $J_1 \cup \cdots \cup J_k \notin \mathcal{B}$.

From this definition, it follows that a subset of a nested set is a nested set. Therefore, the collection of all nested sets of $\mathcal{B}$ forms an abstract simplicial complex with vertex set $\mathcal{B}$, called the *nested set complex* $\triangle_{\mathcal{B}}$ of $\mathcal{B}$. It turns out that the nestohedron $P_{\mathcal{B}}$ is always a simple polytope, and the face lattice of $P_{\mathcal{B}}$ does not depend on the values of the positive parameters y_I, but only on $\mathcal{B}$, and this face lattice is dual to the face lattice of the nested set complex $\triangle_{\mathcal{B}}$; see [56, Theorem 7.4].

Example 8.3.4. The collection $\mathcal{B} = \{\{1\},\{2\},\{3\},\{4\},\{1,2,4\},\{1,3,4\},\{2,3,4\},\{1,2,3,4\}\}$ is a building set on $[4]$. The corresponding nested sets and their dimensions as faces in $\triangle_{\mathcal{B}}$ are as follows:

$$\left\{\begin{array}{lc}
\dim = -1 & \varnothing \\
\hline
\dim = 0 & \{1\},\{2\},\{3\},\{4\},\{124\},\{134\},\{234\} \\
\hline
\dim = 1 & \{1,2\},\{1,3\},\{2,3\},\{1,4\},\{2,4\},\{3,4\},\{1,124\},\{1,134\}, \\
 & \{2,124\},\{2,234\},\{3,134\},\{3,234\},\{4,124\},\{4,134\},\{4,234\} \\
\hline
\dim = 2 & \{1,2,3\},\{1,2,124\},\{1,3,134\},\{2,3,234\},\{1,4,124\}, \\
 & \{1,4,134\},\{2,4,124\},\{2,4,234\},\{3,4,134\},\{3,4,234\}
\end{array}\right\}.$$

This building set's nestohedron is the simple polytope in Figure 8.4. Each of the vertices corresponds to a maximal nested set. In fact, each of the faces of the nestohedron corresponds to the nested set that is the intersection of the maximal nested sets that correspond to the vertices of the face.

There are two special families of vectors $\mathbf{b}$ for which $\overline{\mathfrak{X}}_n(\mathbf{b})$ is a translation by $\mathbf{e}_{n+1}$ of a nestohedron whose face structure can be determined without much more

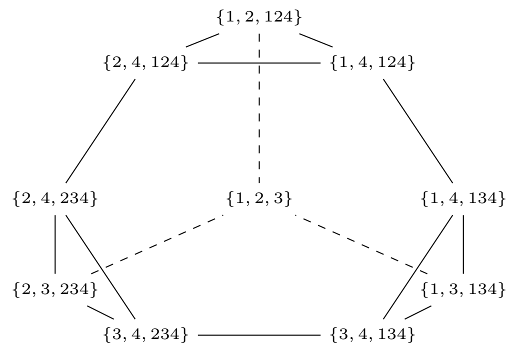


Figure 8.4: The nestohedron from Example 8.3.4.

work.

Definition 8.3.5. A **stellohedron** St_n is a nestohedron for the building set

$$\{\{1\},\{2\},\ldots,\{n\}\}\cup\{S\cup\{n+1\}\}_{S\subseteq[n]}.$$

Proposition 8.3.6. *Let* $\mathbf{b}\in\mathbb{Z}^n_{>0}$*. Let* y_I $(I\subseteq[n+1])$ *be the associated coefficients given in Proposition 8.2.13. Assume that* $y_I>0$ *for all sets* I *for which* $|I|=1$ *and for all sets* I *for which* $n+1\in I$ *and* $|I|\geq 3$.

(a) If $y_I>0$ *for all sets* I *of the form* $I=\{i,n+1\}$ $(i\leq n)$*, then the* $\mathbf{b}$*-parking-function polytope* $\mathfrak{X}_n(\mathbf{b})$ *is combinatorially equivalent to the stellohedron* St_n.

(b) If $y_I=0$ *for all sets* I *of the form* $I=\{i,n+1\}$ $(i\leq n)$*, then* $\mathfrak{X}_n(\mathbf{b})$ *is combinatorially equivalent to the nestohedron for the building set*

$$\{\{1\},\{2\},\ldots,\{n\}\}\cup\{S\cup\{n+1\}\}_{S\subseteq[n],\,|S|\neq 1}.$$

Proof. For Proposition 8.3.6(a), observe that $\mathfrak{X}_n(\mathbf{b})$ is in this case combinatorially equivalent to the following Mikowski sum:

$$\mathfrak{X}_n(\mathbf{b}) \cong \overline{\mathfrak{X}}_n(\mathbf{b}) \cong \sum_{i=1}^{n} \triangle_{\{i\}} + \sum_{S\subseteq[n],\, S\neq\varnothing} \triangle_{S\cup\{n+1\}}.$$

Note that the indices of the standard simplices in the above Minkowski sum range over the collection $\{\{1\},\{2\},\ldots,\{n\}\} \cup \{S \cup \{n+1\}\}_{\varnothing\neq S\subseteq[n]}$. If we add $\{n+1\}$ to this collection, the building set that results corresponds to the stellohedron [55, Sec 10.4]. Adding $\triangle_{\{n+1\}}$ to the Minkowski sum only translates the polytope by $\mathbf{e}_{n+1}$ and does not change the face structure. Hence, we have

$$\mathfrak{X}_n(\mathbf{b}) \cong \sum_{i=1}^{n} \triangle_{\{i\}} + \sum_{S\subseteq[n],\, S\neq\varnothing} \triangle_{S\cup\{n+1\}} + \triangle_{\{n+1\}} \cong \mathsf{St}_n.$$

For Proposition 8.3.6(b), we have in this case that $b_1 = 1$, and so, similarly to the previous case,

$$\begin{aligned}\mathfrak{X}_n(\mathbf{b}) \cong \overline{\mathfrak{X}}_n(\mathbf{b}) &\cong \sum_{i=1}^{n} \triangle_{\{i\}} + \sum_{S\subseteq[n],\, |S|\geq 2} \triangle_{S\cup\{n+1\}} \\ &\cong \sum_{i=1}^{n} \triangle_{\{i\}} + \sum_{S\subseteq[n],\, |S|\geq 2} \triangle_{S\cup\{n+1\}} + \triangle_{\{n+1\}}.\end{aligned}$$

The indices of the standard simplices in the above Minkowski sum range over the collection

$$\{\{1\},\{2\},\ldots,\{n\}\} \cup \{S \cup \{n+1\}\}_{S\subseteq[n],\,|S|\neq 1},$$

which is also a building set. Therefore, $\mathfrak{X}_n(\mathbf{b})$ is combinatorially equivalent to the nestohedron corresponding to this building set. $\square$

We will see later in Corollary 8.3.14 that the two face structures in Proposition 8.3.6 are the only combinatorial types that can occur on $\mathfrak{X}_n(\mathbf{b})$ for *any* $\mathbf{b} \in \mathbb{Z}^n_{>0}$, even if the corresponding y_I do not satisfy the positivity conditions of Proposition 8.3.6. We introduce the notation

$$\mathcal{B}_{\mathsf{St}_n} := \{\{1\},\{2\},\ldots,\{n\}\} \cup \{S \cup \{n+1\}\}_{S\subseteq[n]},$$
$$\mathcal{B}_{\mathsf{PF}_n} := \{\{1\},\{2\},\ldots,\{n\}\} \cup \{S \cup \{n+1\}\}_{S\subseteq[n],\,|S|\neq 1},$$

for the two building sets in Proposition 8.3.6. We also introduce some terminology from [16] to characterize the nested sets. We say a subset $I \subseteq [n]$ and a (nonempty) chain $\mathcal{F} = (A_1 \subsetneq \cdots \subsetneq A_k)$ in the Boolean lattice $(2^{[n]}, \subseteq)$ are *compatible*, denoted by $I \leq \mathcal{F}$, if $I \subseteq \min(\mathcal{F}) = A_1$. If $\mathcal{F}$ is the empty chain $\varnothing$, then we consider every subset I to be compatible with $\varnothing$. The nested sets for $\mathcal{B}_{\mathsf{St}_n}$ and $\mathcal{B}_{\mathsf{PF}_n}$ are characterized in the next propositions.

Proposition 8.3.7 ([49, Prop. 3.7])**.** *The nested sets for $\mathcal{B}_{\mathsf{St}_n}$ have form $\{\{i\}\}_{i\in I} \cup \{A \cup \{n+1\}\}_{A\in\mathcal{F}}$, where $I \subseteq [n]$ and $\mathcal{F} \subset 2^{[n]} - \{[n]\}$ is a chain such that $I \leq \mathcal{F}$ is a compatible pair.*

The building set and the collection of nested sets in Example 8.3.4 is the case of $n = 3$ of the proposition below.

Proposition 8.3.8. *The nested set for $\mathcal{B}_{\mathsf{PF}_n}$ is of the form $\{\{i\}\}_{i\in I} \cup \{A\}_{A\in\mathcal{F}}$, where $I \subseteq [n+1]$ and $|I| \leq 2$ if $n+1 \in I$, and $\mathcal{F} \subset 2^{[n+1]} - \{[n+1]\}$ is a chain such that $|\min(\mathcal{F})| \geq 3$ and $n+1 \in \min(\mathcal{F})$, and $I \leq \mathcal{F}$ is a compatible pair.*

Proof. Let N be a nested set for $\mathcal{B}_{\mathsf{PF}_n}$. Let I be the subset consisting of $i \in [n+1]$ for any singleton $\{i\} \in N$ and $\mathcal{F}$ be the collection of all subsets in N that are not a singleton. By the definition of nested set for $\mathcal{B}_{\mathsf{PF}_n}$, we must have $n+1 \in A$ and $|A| \geq 3$ for any $A \in \mathcal{F}$. Pick any $A_1, A_2 \in \mathcal{F}$, we always have $n+1 \in A_1 \cap A_2 \neq \varnothing$. Hence one of A_1 or A_2 must be a subset of the other and therefore $\mathcal{F}$ is a chain. For any $i \in I$, we must have $i \in A$ for all $A \in \mathcal{F}$; otherwise, there exists $A' \in \mathcal{F}$ such that $\{i\}$ and A' are disjoint subsets in N with $A' \cup \{i\} \in \mathcal{B}_{\mathsf{PF}_n}$, which violates the condition (N2) in the definition of nested sets. This implies $I \subseteq \min(\mathcal{F})$, so $I \leq \mathcal{F}$. If $n+1 \in I$ and $|I| \geq 3$, then the union of all singletons in I, which is I itself, is in $\mathcal{B}_{PF_n}$, which again violates condition (N2). Thus, we have $|I| \leq 2$ if $n+1 \in I$. Conversely, it is routine to check that such a collection $\{\{i\}\}_{i\in I} \cup \{A\}_{A\in\mathcal{F}}$ satisfies the two axioms of the definition of nested sets. $\square$

In order to understand the face lattice of $\mathfrak{X}_n(\mathbf{b})$ in terms of building sets, we now generalize the notation U_i for certain facets of $\mathfrak{X}_n(\mathbf{b})$ introduced immediately before Lemma 8.2.5 by setting

$$U_I := \left\{(x_1,\dots,x_n) \in \mathfrak{X}_n(\mathbf{b}) \,:\, \sum_{i\in I} x_i = \sum_{i=n-|I|+1} S_i\right\},$$

for all nonempty $I \subseteq [n]$, except in the case $|I| = n-1$ if $b_1 = 1$. Thus, U_I is the facet of $\mathfrak{X}_n(\mathbf{b})$ defined by the ($\mathbf{b}$-parking) inequality for the subset I. In terms of this notation, the facets U_j defined prior to Lemma 8.2.5 are given by $U_j = U_{I_j}$, where $I_j := \{j, j+1, \dots, n\}$. Comparing the supporting hyperplanes of the facets of $\mathfrak{X}_n(\mathbf{b})$ in Theorem 8.2.3 and the two building sets in Proposition 8.3.6, we immediately arrive to the following lemma.

Lemma 8.3.9. *Let $\mathcal{B} := \mathcal{B}_{\mathsf{PF}_n}$ if $b_1 = 1$, and let $\mathcal{B} := \mathcal{B}_{\mathsf{St}_n}$ if $b_1 \geq 2$. The facets of $\mathfrak{X}_n(\mathbf{b})$ are in bijection with the elements of $\mathcal{B} \setminus \mathcal{B}_{max} = \mathcal{B} \setminus \{[n+1]\}$. If $b_1 = 1$, then the bijection is as follows:*

$$\begin{aligned} L_i &\longleftrightarrow \{i\} && \textit{for } 1 \leq i \leq n, \\ U_{[n]\setminus S} &\longleftrightarrow S \cup \{n+1\}, && \textit{for } S \subsetneq [n],\ |S| \neq 1. \end{aligned}$$

If $b_1 \geq 2$, then the bijection is as follows:

$$\begin{aligned} L_i &\longleftrightarrow \{i\} && \textit{for } 1 \leq i \leq n, \\ U_{[n]\setminus S} &\longleftrightarrow S \cup \{n+1\}, && \textit{for } S \subsetneq [n]. \end{aligned}$$

For any $B \in \mathcal{B} \setminus \{[n+1]\}$, we denote by F_B the corresponding facet of $\mathfrak{X}_n(\mathbf{b})$ under the bijection in Lemma 8.3.9.

Lemma 8.3.10. *Let $\mathcal{B} \coloneqq \mathcal{B}_{\mathsf{PF}_n}$ if $b_1 = 1$, and let $\mathcal{B} \coloneqq \mathcal{B}_{\mathsf{St}_n}$ if $b_1 \geq 2$. Then the map*

$$N \longmapsto \bigcap_{B \in N} F_B$$

*is a bijection between the maximal nested sets N (*i.e.*, the facets of $\triangle_{\mathcal{B}}$) and the vertices of $\mathfrak{X}_n(\mathbf{b})$.*

Proof. Consider $b_1 = 1$, the maximal nested sets for $\mathcal{B}_{\mathsf{PF}_n}$ are of the form $\{\{i\}\}_{i \in I} \cup \{A_1 \subsetneq A_2 \subsetneq \ldots \subsetneq A_{n-|I|}\}$ satisfying the condition of Proposition 8.3.8 and $|A_{i+1}| - |A_i| = 1$ for all i. If $n+1 \notin I$, then the corresponding vertex can be obtained as a vector in $\mathbb{R}^n$ by placing 1s in positions I, then placing S_n in position $[n+1] - A_{n-|I|}$, placing S_{n-1} in position $A_{n-|I|} - A_{n-|I|-1}$, and so on, in the final step placing $S_{|I|+1}$ in position $A_2 - A_1$. In this way, we fill out all n entries of the vector in $\mathbb{R}^n$, and the resulting vector is the permutation of $\mathbf{v}_{|I|}$ which is a vertex by Theorem 8.2.3(b). If $n+1 \in I$, then $I = \{\{i\}, \{n+1\}\}$ for some $i \in [n]$. Then we obtain the vertex as a vector in $\mathbb{R}^n$ by placing $1 = b_1 = S_1$ in position i, placing S_n in position $[n+1] - A_{n-2}$, placing S_{n-1} in position $A_{n-2} - A_{n-3}$, etc., placing S_3 in position $A_2 - A_1$, in the final step placing S_2 in position $A_1 - \{i, n+1\}$. In this case, the resulting vector is the permutation of $\mathbf{v}_1 = \mathbf{v}_0$ which is also a vertex by Theorem 8.2.3(b). As a result, this gives a bijection between the maximal nested set for $\mathcal{B}_{\mathsf{PF}_n}$ and the vertices of

$\mathfrak{X}_n(\mathbf{b})$ for $b_1 = 1$ via the intersections of facets corresponding to the maximal nested sets.

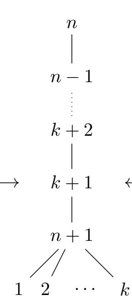
For the case of $b_1 \geq 2$, the proof is similar. $\square$

Remark 8.3.11. In both cases $b_1 = 1$ and $b_1 \geq 2$, one can further consider the vertex posets Q_v for every v in the set of vertices $\mathrm{Vert}(\mathfrak{X}_n(\mathbf{b}))$ (see Subsection 8.2.3) cooperating with the bijection in Lemma 8.3.10. It gives the following commutative diagram

$$\begin{tikzcd}
\mathrm{Vert}(\mathfrak{X}_n(\mathbf{b})) \arrow[rr] \arrow[dr] & & \{\text{maximal nested sets for } \mathcal{B}\} \\
& \{Q_v : v \in \mathrm{Vert}(\mathfrak{X}_n(\mathbf{b}))\}. \arrow[ur] &
\end{tikzcd}$$

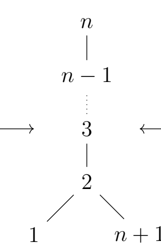
For example, the following are the correspondences between the weakly increasing vertices of $\mathfrak{X}_n(\mathbf{b})$ for $b_1 = 1$ and their vertex posets and the maximal nested sets:

$$\Big(\underbrace{1,\dots,1}_{k}, S_{k+1}, \dots, S_n\Big) \longleftrightarrow$$



$$\longleftrightarrow \left\{\begin{matrix} \{1\},\ \{2\},\dots,\{k\}, \\ [k]\cup\{n+1\}, \\ [k+1]\cup\{n+1\}, \\ \dots, \\ [n-1]\cup\{n+1\} \end{matrix}\right\}$$

$$(1, S_2, \dots, S_n) \longleftrightarrow$$



$$\longleftrightarrow \left\{\begin{matrix} \{1\},\ \{n+1\}, \\ [2]\cup\{n+1\}, \\ [3]\cup\{n+1\}, \\ \dots, \\ [n-1]\cup\{n+1\} \end{matrix}\right\}.$$

With the aid of Lemma 8.3.9 and Lemma 8.3.10, we can completely characterize the face structure of the $\mathbf{b}$-parking function polytope for any $\mathbf{b}$ as follows.

Theorem 8.3.12. *Let $\mathcal{B} \coloneqq \mathcal{B}_{\mathsf{PF}_n}$ if $b_1 = 1$, and let $\mathcal{B} \coloneqq \mathcal{B}_{\mathsf{St}_n}$ if $b_2 \geq 1$. Then the map*

$$\triangle_{\mathcal{B}} \longrightarrow \{\textit{nonempty faces of } \mathfrak{X}_n(\mathbf{b})\}, \qquad \textit{where} \qquad N \longmapsto \mathcal{F}_N \coloneqq \bigcap_{B \in N} F_B$$

is an isomorphism between the dual of the face lattice of the nested set complex $\triangle_{\mathcal{B}}$ *and the face lattice of* $\mathfrak{X}_n(\mathbf{b})$*.*

Proof. Consider $\mathbf{b} = (b_1, \ldots, b_n)$ with $b_1 = 1$. We first prove that the map from the nested sets in $\triangle_{\mathcal{B}_{\mathsf{PF}_n}}$ to the set of nonempty faces of $\mathfrak{X}_n(\mathbf{b})$ is well-defined. From Lemma 8.3.9, the singleton nested set $\{B\} \in \triangle_{\mathcal{B}_{\mathsf{PF}_n}}$ corresponds to the facet F_B of $\mathfrak{X}_n(\mathbf{b})$. Since any intersection of facets is a face of the polytope, we know $\mathcal{F}_N$ is a face of $\mathfrak{X}_n(\mathbf{b})$ for any nested set $N \in \triangle_{\mathcal{B}_{\mathsf{PF}_n}}$. For any nested set N, the face $\mathcal{F}_N$ is nonempty by Lemma 8.3.10 because $\mathcal{F}_N$ contains those vertices that correspond to the maximal nested sets containing N. Hence, the map is well-defined.

Now we check the injectivity of the map. By Theorem 8.2.3(a), $\mathfrak{X}_n(\mathbf{b})$ is a simple polytope. In a simple polytope, every nonempty intersection of facets determines a unique face. Hence, the injectivity of the map follows.

We now show that the map is surjective. Let N' be a subset of $\mathcal{B}_{\mathsf{PF}_n}$ such that the face $\bigcap_{B \in N'} F_B$ is nonempty. Say $N' = \{\{i\}\}_{i \in I'} \cup \{A\}_{A \in \mathcal{F}'}$ for some $I' \subset [n+1]$ and $\mathcal{F}'$ consisting of non-singleton subsets in $\mathcal{B}_{\mathsf{PF}_n}$, then

$$\bigcap_{B \in N'} F_B = \left(\bigcap_{i \in I'} L_i \right) \cap \left(\bigcap_{A \in \mathcal{F}'} U_{[n] \setminus A} \right).$$

We now show that N' must be a nested set for $\mathcal{B}_{\mathsf{PF}_n}$ (Recall Proposition 8.3.8).

- By definition of $\mathcal{B}_{\mathsf{PF}_n}$, we have $n+1 \in A$ and $|A| \geq 3$ for any $A \in \mathcal{F}'$ right away.
- First, for any $A \in \mathcal{F}'$ we must have $I' \subset A$; otherwise, there is $j \in I'$ but $j \notin A$ for some $A \in \mathcal{F}'$. Then this face $\bigcap_{B \in N'} F_B$ is contained in the intersection of the facets $F_j \cap U_{[n] \setminus A}$, which is an empty set since $j \in [n] \setminus A$. One can check the

defining equations of the hyperplanes that there are no vertices of $\mathfrak{X}_n(\mathbf{b})$ in this intersection. This contradicts the assumption that the face is nonempty.

- Second, we claim if $n+1 \in I'$, then $|I'| \leq 2$. This is because the face $\bigcap_{B \in N'} F_B$ is contained in the facet $U_{[n]}$ whose supporting hyperplane is $\sum_{i=1}^{n} x_i = S_n + S_{n-1} + \ldots + S_1$. Any vertex of $\mathfrak{X}_n(\mathbf{b})$ on $U_{[n]}$ is the permutation of $(S_1, S_2, \ldots, S_n) = (1, S_2, \ldots, S_n)$ which has exactly one coordinate being 1. Thus, I' does not contain more than one element that is not $n+1$.

- In the final step, we show that $\mathcal{F}'$ must be a chain. Let $A_1, A_2 \in \mathcal{F}'$ with $|A_1| \leq |A_2|$ then we must have $A_1 \subseteq A_2$. If not, there exists $j \in A_1$, but $j \notin A_2$. Notice that any vertex $v = (v_1, \ldots, v_n)$ of $\mathfrak{X}_n(\mathbf{b})$ on the facet $U_{[n]\setminus A_2}$ has coordinates satisfying

$$\{v_i\}_{i \in [n]\setminus A_2} = \{S_n, S_{n-1}, \ldots, S_{|A_2|+1}\};$$

while the vertex v on $U_{[n]\setminus A_1}$ has coordinates satisfying

$$\{v_i\}_{i \in [n]\setminus A_1} = \{S_n, \ldots, S_{|A_2|+1}, \ldots, S_{|A_1|+1}\}.$$

Then there is no vertex of $\mathfrak{X}_n(\mathbf{b})$ that lies on $U_{[n]\setminus A_2}$ and $U_{[n]\setminus A_1}$ simultaneously since all the vertices have at most one coordinate equal to any one of $S_n, S_{n-1}, \ldots, S_{|A_2|+1}$.

Therefore, by Proposition 8.3.8 the subset N' is a nested set for $\mathcal{B}_{\mathsf{PF}_n}$.

Now the map is a bijection. By the definition of the map, it is clear that any two nested sets $N_1 \subseteq N_2$ if and only if $\mathcal{F}_{N_1} \supseteq \mathcal{F}_{N_2}$. Therefore, the map is a poset isomorphism. For the case that $b_1 \geq 2$, the proof is similar. $\square$

Recall that given a building set $\mathcal{B} \subset 2^{[n]}$, the face lattice of the nestohedron

$P_{\mathcal{B}} = \sum_{I\in\mathcal{B}} y_I \triangle_I$ (where $y_I > 0$) is completely determined by the building set $\mathcal{B}$ and is dual to the face lattice of the nested set complex $\triangle_{\mathcal{B}}$. It is somewhat surprising that the face lattice of $\mathfrak{X}_n(\mathbf{b})$ is determined by the two building sets $\mathcal{B}_{PF_n}$ and $\mathcal{B}_{\mathsf{St}_n}$ because from Proposition 8.2.13 we have seen that for most of b the polytope $\overline{\mathfrak{X}}_n(\mathbf{b})$ is not a nestohedron (or a translation of one). It will be interesting to understand how the value of y_I affects the face structure of $\sum_I y_I \triangle_I$. More concretely, can we characterize the parameters $\{y_I\}$ such that $\Sigma y_I \triangle_I$ is combinatorially equivalent to $P_{\mathcal{B}}$ for a given building set $\mathcal{B}$?

From Theorem 8.3.12, we can explicitly tell which facets intersect at a given face and all the vertices on this face by observing the nested set corresponding to the face; we can see this exemplified next.

Example 8.3.13. For $\mathbf{b} = (1, b_2, \ldots, b_6)$, consider the nested set $N = \{\{1\}, \{3\}\} \cup (\{1, 3, 4, 7\} \subsetneq \{1, 2, 3, 4, 7\})$. Then the face F_N is the intersection of the following facets of $\mathfrak{X}_6(\mathbf{b})$:

$$L_1 : x_1 = 1, L_3 : x_3 = 1, U_{\{1,3,4\}} : x_2+x_5+x_6 = S_6+S_5+S_4, U_{\{1,2,3,4\}} : x_5+x_6 = S_6+S_5.$$

The face F_N is 2-dimensional and has 4 vertices:

$$(1, S_4, 1, 1, S_5, S_6), (1, S_4, 1, S_3, S_5, S_6), (1, S_4, 1, 1, S_6, S_5), (1, S_4, 1, S_3, S_6, S_5).$$

Corollary 8.3.14. *For* any $\mathbf{b} = (b_1, b_2, \ldots, b_n) \in \mathbb{Z}^n_{>0}$, *the* $\mathbf{b}$*-parking-function polytope* $\mathfrak{X}_n(\mathbf{b})$ *is combinatorially equivalent to the classical parking-function polytope* PF_n *if* $b_1 = 1$ *or is combinatorially equivalent to the stellohedron* St_n *if* $b_1 \geq 2$.

Next, we provide a visualization of the two 3-dimensional cases of Corollary 8.3.14.

Example 8.3.15. The combinatorial type of $\mathfrak{X}_n(\mathbf{b})$ is as in Figure 8.4 when $n = 3$

and $b_1 = 1$ and is as in Figure 8.5 when $n = 3$ and $b_1 \geq 2$. The coordinate of each vertex can be obtained by using the bijection described in the proof of Lemma 8.3.10.

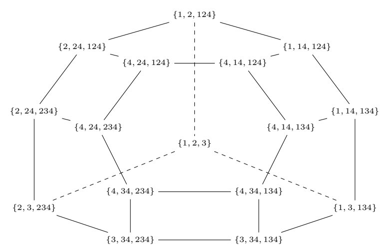


Figure 8.5: The combinatorial type of $\mathfrak{X}_n(\mathbf{b})$ when $n = 3$ and $b_1 \geq 2$.

Up to this point we have observed $\mathfrak{X}_n(\mathbf{b})$ as a generalized permutahedron and have characterized its combinatorial types through the use of building sets. In the next section, we exhibit $\mathfrak{X}_n(\mathbf{b})$ as a special polymatroid, which allows us to obtain further results.

8.4 A polymatroid perspective

Here we study the **b**-parking-function polytope through a polymatroid lens. Polymatroids are a class of polyhedra that were introduced by Edmonds [27] in 1970 as polyhedral generalizations of matroids. We note that generalized permutahedra are equivalent (up to translation) to polymatroids. By exhibiting $\mathfrak{X}_n(\mathbf{b})$ as a polymatroid, we are able to draw conclusions about properties such as its combinatorial and circuit diameters. For background on diameters, see [43, 73] and [11, 12, 15, 28]. We begin by recalling a few helpful terms and definitions from matroid theory [53].

Definition 8.4.1. Let $f : 2^{[n]} \to \mathbb{R}$ be a set function. The function f is said to be **submodular** if for every $X, Y \subseteq [n]$ with $X \subseteq Y$ and every $z \in [n] \setminus Y$, the following inequality holds: $f(X \cup \{z\}) - f(X) \geq f(Y \cup \{z\}) - f(Y)$. The *polymatroid* associated with f is the polytope

$$P_f \coloneqq \left\{ x \in \mathbb{R}^n : x \geq 0,\ \sum_{i \in S} x_i \leq f(S) \text{ for all } S \subseteq [n] \right\}.$$

A set function $f\colon 2^{[n]} \to \mathbb{R}$ is said to be *nondecreasing* if, for every $X, Y \subseteq [n]$ with $X \subseteq Y$, we have $f(X) \leq f(Y)$. Henceforth, let $f_{\mathbf{b}}\colon 2^{[n]} \to \mathbb{Z}_+$ be the set function defined by setting $f_{\mathbf{b}}(\varnothing) \coloneqq 0$ and

$$f_{\mathbf{b}}(I) \coloneqq -|I| + \sum_{j=1}^{n} \min\{|I|, j\}\, b_{n-j+1}, \qquad \text{for } \varnothing \neq I \subseteq [n].$$

Proposition 8.4.2. *The function $f_{\mathbf{b}}$ is a nondecreasing submodular function.*

Proof. By construction, $f_{\mathbf{b}}$ is integer and nondecreasing. It remains to show that $f_{\mathbf{b}}$ is submodular. Observe that $f_{\mathbf{b}}(I) = -|I| + \sum_{j=1}^{n} \min\{|I|, j\}\, b_{n-j+1}$ only depends on the cardinality $|I|$ of I, and not on the specific elements in I. It suffices to prove that $I \subset J \subsetneq [n] \Rightarrow f_{\mathbf{b}}(|I|+1) - f_{\mathbf{b}}(|I|) \geq f_{\mathbf{b}}(|J|+1) - f_{\mathbf{b}}(|J|)$, or equivalently

$$i \leq j < n \quad \Rightarrow \quad f_{\mathbf{b}}(i+1) - f_{\mathbf{b}}(i) \geq f_{\mathbf{b}}(j+1) - f_{\mathbf{b}}(j).$$

The difference $f_{\mathbf{b}}(i+1) - f_{\mathbf{b}}(i)$ is

$$f_{\mathbf{b}}(i+1) - f_{\mathbf{b}}(i) = -(i+1) - (-i) + \sum_{k=1}^{n} (\min\{i+1, k\} - \min\{i, k\})\, b_{n-k+1} = -1 + \sum_{k=i+1}^{n} b_{n-k+1}.$$

As $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{Z}^n_{>0}$,

$$\sum_{k=i+1}^{n} b_{n-k+1} \geq \sum_{k=j+1}^{n} b_{n-k+1} \qquad \text{for} \quad i \leq j.$$

Thus, $f_{\mathbf{b}}$ is submodular. $\square$

We translate the inequality description of $\mathfrak{X}_n(\mathbf{b})$ (Theorem 8.2.3(c)) to a description of the form P_f. Setting $y_i \coloneqq x_i - 1$, one obtains

$$x_i \geq 1 \quad \Leftrightarrow \quad x_i - 1 \geq 0 \quad \Leftrightarrow \quad y_i \geq 0,$$

and the remaining constraints in Theorem 8.2.3(c) take the form

$$\sum_{i \in I} x_i \leq \sum_{j=1}^{n} \min\{|I|, j\}\, b_{n-j+1} \quad \Leftrightarrow \quad \sum_{i \in I} y_i \leq -|I| + \sum_{j=1}^{n} \min\{|I|, j\}\, b_{n-j+1}.$$

The resulting polymatroid is now described as

$$\mathfrak{X}'_n(\mathbf{b}) \coloneqq \left\{ y \in \mathbb{R}^n \,:\, y \geq 0, \sum_{i \in I} y_i \leq f_{\mathbf{b}}(I) \text{ for all } I \subseteq [n] \right\},$$

which proves the following.

Theorem 8.4.3. *The polytope $\mathfrak{X}'_n(\mathbf{b})$ is a polymatroid. In fact, $\mathfrak{X}'_n(\mathbf{b})$ is a special polymatroid for which upper bounds on the sum of entries in any set I depend only on the cardinality $|I|$.*

The polytopes $\mathfrak{X}_n(\mathbf{b})$ and $\mathfrak{X}'_n(\mathbf{b})$ are combinatorially equivalent. Thus, we can leverage the theory of polymatroids to obtain results about $\mathfrak{X}_n(\mathbf{b})$. In particular, we note that known results about polymatroids allow for alternate directions to recover Theorem 8.2.3. Topkis [65] studied extreme point adjacency in polymatroids [65, Corollary 5.4], and his work can be used to deduce Theorem 8.2.3(a). Shapley's observation on the structure of the vertices of a polymatroid [61] can be used to obtain Theorem 8.2.3(b). Edmonds [27, 59] showed that we can optimize linear functions over polymatroids in strongly polynomial by certifying *total dual integrality* through the greedy algorithm. In turn, this yields another direction for proving the inequality description in Theorem 8.2.3(c).

Next, using Theorem 8.4.3 and Proposition 8.2.6, we can carry over known upper bounds on the combinatorial diameters of polymatroids [66] to $\mathfrak{X}_n(\mathbf{b})$, of the form $\min\{2n, \frac{1}{2}n(n-1)+1\}$. Recall that the *combinatorial diameter* of a polytope is the diameter of its edge graph, that is, the maximum length of a shortest edge-path between two vertices. We now provide a direct proof of these upper bounds, which allows us to improve them for the case $b_1 = 1$ and see tightness in all cases.

Theorem 8.4.4. *If $b_1 > 1$, the combinatorial diameter of $\mathfrak{X}_n(\mathbf{b})$ is exactly $\min\{2n, \frac{1}{2}n(n-1)+1\}$. If $b_1 = 1$, the combinatorial diameter is exactly $\min\{2(n-1), \frac{1}{2}n(n-1)\}$.*

Proof. Note that $2n \leq \frac{1}{2}n(n-1)+1$ and $2(n-1) < \frac{1}{2}n(n-1)$ if and only if $n \geq 5$. For $n \leq 4$, tightness of the bounds can be verified exhaustively. For example, for $n = 3$, it is clear from Figure 8.1 that the combinatorial diameter of $\mathfrak{X}_3(1,2,3)$ and $\mathfrak{X}_3(2,3,4)$ is 3 and 4, respectively.

We provide an argument for the bounds $2n$ and $2(n-1)$. Let $\mathbf{w}_k = \pi(\mathbf{v}_k)$ be a vertex of $\mathfrak{X}_n(\mathbf{b})$ where $k < n$. By Proposition 8.2.6, $\mathbf{w}_k$ is adjacent to $\mathbf{w}_{k+1} = \pi(\mathbf{v}_{k+1})$, which corresponds to reducing the $(k+1)^{\text{th}}$ entry of $\mathbf{v}_k$ to 1. Consider the path $P_k = (\mathbf{w}_k, \mathbf{w}_{k+1}, \ldots, \mathbf{w}_{n-1}, \mathbf{1})$ on $\mathfrak{X}_n(\mathbf{b})$, where P_k ends at $\mathbf{v}_n = \mathbf{w}_n = \mathbf{1}$. Because P_k begins at $\mathbf{w}_k$, the length of P_k is at most n when $b_1 > 1$. Concatenating paths when necessary, we have a path of length at most $2n$ between any two vertices of $\mathfrak{X}_n(\mathbf{b})$ with $b_1 > 1$. A path of this length is realized when traveling between the vertices $\mathbf{w}_0 = (b_1, S_2, \ldots, S_n)$ and $\pi'(\mathbf{w}_0) = (S_n, S_{n-1}, \ldots, S_2, b_1)$. When $b_1 = 1$, $\mathbf{w}_0 = \mathbf{w}_1$, so P_0 has length at most $n-1$. A path of length $2(n-1)$ is realized when travelling between vertices $\mathbf{w}_1 = (1, S_2, \ldots, S_n)$ and $\pi'(\mathbf{w}_1) = (S_n, S_{n-1}, \ldots, S_2, 1)$.

It remains to show tightness. We exhibit that the paths constructed above are shortest paths between the given pairs of vertices. The bound $2n$ for a path between

$\mathbf{w}_0$ and $\pi'(\mathbf{w}_0)$ is achieved through a combination of n reductions and n increases of entries S_i, and no swaps. A path that exclusively uses swaps requires at least $\frac{1}{2}n(n-1)$ steps, and so is not better. Let S_j be the largest value that is reduced during an edge walk. It can be seen that any walk where $j = 1$ has to perform the same number of swaps as one exclusively using swaps; any walk where $j = n-1$ has to perform at least two further steps (in addition to $n-1$ reductions and $n-1$ increases). Let now $2 \leq j \leq n-2$. Then the number of swaps required to arrive at a vertex where the j largest values $S_n, \ldots, S_{n-j+1}$ are in the target positions is bounded below by $j \cdot (n-j)$. (For every $1 \leq i_1 < i_2 \leq n$, it takes $i_2 - i_1$ swaps to move S_{i_2} to the position in which S_{i_1} lies; in a best case, the target positions for $S_n, \ldots, S_{n-j+1}$ start these swaps with values $S_j, \ldots, S_1$.) Combined with the at least j reductions and j increases, the path has length at least $2j + j(n-j) \geq 2j + 2(n-2) \geq 2n$.

An analogous argument with $n-1$ in place of n shows tightness of $2(n-1)$ for $b_1 = 1$. □

Circuits and circuit diameters generalize the concepts of edges and combinatorial diameters. We refer the reader to [11, 12, 21, 28, 57] for background. The set of *elementary vectors* or *circuits* $\mathcal{C}(P)$ of a polyhedron P correspond to the inclusion-minimal dependence relations in the constraint matrices of a (minimal) representation. Note that the elementary vectors or circuits of a polymatroid are *not* the circuits of the underlying matroid. The elementary vectors include all edge directions that appear for varying right-hand sides of a representation, and so $\mathcal{C}(\mathfrak{X}_n(\mathbf{b}))$ contains a superset of the edge directions found in Proposition 8.2.6.

Lemma 8.4.5. *The elementary vectors of $\mathfrak{X}_n(\mathbf{b})$ include $\pm\mathbf{e}_i$ and $\mathbf{e}_i - \mathbf{e}_j$ for all $i, j \leq n$ with $i \neq j$.*

Proof. It suffices to exhibit that for any vector of type $\pm\mathbf{e}_i$ or $\mathbf{e}_i - \mathbf{e}_j$ for all $i, j \leq n$ with $i \neq j$, there exists an $\mathfrak{X}_n(\mathbf{b})$ in which that vector appears as an edge. In fact, they all appear in the same polymatroid: a matroid polyhedron $\mathfrak{X}_n(\mathbf{b})$ for a uniform matroid with rank ≥ 2; *c.f.* [66]. □

The associated *circuit walks* begin and end at vertices, and take maximal steps along elementary vectors instead of edges, in particular through the interior of a polyhedron. The study of circuit diameters is motivated by the search for lower bounds on the combinatorial diameter and the efficiency of circuit augmentation schemes [21]. The *circuit diameter* of a polytope is the maximum length of a shortest circuit walk between any two vertices. Of particular interest are polyhedral families where there is a significant gap between the combinatorial and circuit diameter [13,41]. We will prove that $\mathbf{b}$-parking-function polytopes is one of these families.

Unlike the more technical conditions for the adjacency of vertices found in Proposition 8.2.6, steps along the elementary vectors of Lemma 8.4.5 allow for more general changes to a $\mathbf{b}$-parking function than classic matroid operations: *any* two entries can be swapped (a step in direction $\pm(\mathbf{e}_i - \mathbf{e}_j)$), even for non-bases and even including an entry that is 1; this contrasts with the basis-exchange property of matroids where specific pairs of elements of two bases can be exchanged. Further, *any* entry can be decreased or increased (a step in direction $\pm\mathbf{e}_i$), respectively. The steps along the elementary vectors have to be of maximal length while remaining feasible. Recall from the proof of Proposition 8.4.2 that $\mathfrak{X}_n(\mathbf{b})$ is a special polymatroid for which $f_{\mathbf{b}}(I)$ only depends on the cardinality $|I|$. Thus, any maximal step in direction $\pm(\mathbf{e}_i - \mathbf{e}_j)$ indeed is a swap of the corresponding entries. A maximal step in direction $-\mathbf{e}_i$ corresponds to a decrease of an entry to 1. Note that such a step starting at $\pi(\mathbf{v}_k)$ arrives at a non-vertex when decreasing entry S_i to 1 for $i > k+1$. A maximal step in direction $\mathbf{e}_i$ corresponds to an increase of an entry to the largest S_i that does not appear yet.

We are now ready to prove a bound on the circuit diameter of $\mathfrak{X}_n(\mathbf{b})$.

Theorem 8.4.6. *If $b_1 = 1$, the circuit diameter of $\mathfrak{X}_n(\mathbf{b})$ is at most n. If $b_1 \geq 2$, the circuit diameter is at most $n-1$. Moreover, these bounds are tight when restricted to the circuits of Lemma 8.4.5.*

Proof. Recall that each vertex of $\mathfrak{X}_n(\mathbf{b})$ is a permutation of a point of the form $\mathbf{v}_k$ for $0 \leq k \leq n$. If $b_1 = 1$, then $k \geq 1$. Consider vertices $\pi_1(\mathbf{v}_{k_1})$ and $\pi_2(\mathbf{v}_{k_2})$ for two permutations π_1, π_2 and $k_1 \leq k_2$. Note that $\pi_1(y_{k_1})$ and $\pi_2(y_{k_2})$ have at least k_1 entries 1 and share the largest $n-k_2$ values of y_{k_2} (permuted to different entries). Circuit walks are not reversible, so we will explain how to walk from $\pi_1(\mathbf{v}_{k_1})$ to $\pi_2(\mathbf{v}_{k_2})$ and how to walk from $\pi_2(\mathbf{v}_{k_2})$ to $\pi_1(\mathbf{v}_{k_1})$ in at most the claimed number of steps.

For the walk from $\pi_1(\mathbf{v}_{k_1})$ to $\pi_2(\mathbf{v}_{k_2})$, order the entries in descending order by value in $\pi_2(\mathbf{v}_{k_2})$. In the first $n-k_2$ iterations, a swap of two entries can guarantee that the largest $n-k_2$ entries match. Then follow k_2-k_1 iterations in which entries are decreased to 1. The remaining k_1 entries are already identical to 1. We obtain a circuit walk of at most $n-k_1$ steps.

For the walk from $\pi_2(\mathbf{v}_{k_2})$ to $\pi_1(\mathbf{v}_{k_1})$, order the entries in descending order by value in $\pi_1(\mathbf{v}_{k_1})$. In the first $n-k_2$ iterations, a swap of two entries can guarantee that the largest $n-k_2$ entries match. Then follow k_2-k_1 iterations in which a 1-entry is increased to the next-largest entry in $\pi_1(\mathbf{v}_{k_1})$. The remaining k_1 entries are already identical to 1. Again, we obtain a circuit walk of at most $n-k_1$ steps.

It remains to show that the given bounds are tight if one is restricted to the elementary vectors in Lemma 8.4.5. To this end, observe that any elementary vector can only increase one entry. If $b_1 > 1$, the case where $0 = k_1 \leq k_2 = n$ is possible, and all n entries need to be increased in a circuit walk from $\pi_1(y_{k_1})$ to $\pi_2(y_{k_2})$. If $b_1 > 1$ then $k_1 \geq 1$. In the case where $1 = k_1 \leq k_2 = n$, $n-1$ entries need to be increased. This proves the claim. □

For all $n \geq 2$, the bounds of Theorem 8.4.6 are strictly lower than the bounds for the combinatorial diameter of Theorem 8.4.4. Note that we only worked with a subset of $\mathcal{C}(\mathfrak{X}_n(\mathbf{b}))$. A complete characterization of the whole set $\mathcal{C}(\mathfrak{X}_n(\mathbf{b}))$ remains open. Due to this restriction, we constructed an *integral* circuit walk [15], where each intermediate point is integral. We believe that our bounds are tight if restricted to integral walks and leave this as an open question.

8.5 Further perspectives

Recall that the classical parking-function polytope PF_n is the convex hull of all parking functions of length n in $\mathbb{R}^n$. As noted in [36, Remark 2.10] and [9, Remark 4.8], there are other alternative perspectives on PF_n. It can be viewed as:

- a special case of the main object of study for this paper, the **b**-parking-function polytope when $\mathbf{b} = (1, 1, 1, \ldots, 1)$,
- a particular *partial permutahedron* $\mathcal{P}(n, n-1)$ as defined in [37] and further studied in [9],
- a specific *polytope of win vectors* as defined by Bartels et al. in [7] (when specializing

to the complete graph, the work of Backman [3, Theorem 4.5 and Corollary 4.6] provides alternate routes for computing the volume and lattice-point enumerator of PF_n), and

- essentially as the stochastic sandpile model for the complete graph [60].

In addition to these interpretations, as direct corollaries to Theorem 8.2.11, Theorem 8.3.12, and Theorem 8.4.3, we can also add that (a linear projection of) PF_n:

- is a special generalized permutahedron,
- can be seen as a building set, and
- is a special type of polymatroid, respectively.

We include two detailed interpretations of the classical parking-function polytope as a projection of a relaxed Birkhoff or assignment polytope, and as a projection of a relaxed partition polytope.

8.5.1 Relation of PF_n to Birkhoff/assignment polytopes

Consider an assignment problem for n cars and n parking spots, represented with decision variables $x_{ij} \in \{0, 1\}$ indicating car i being assigned to spot j. The feasible set is a *Birkhoff polytope* or *assignment polytope* as follows [5]:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } 1 \le i \le n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for all } 1 \leq j \leq n$$

$$x_{ij} \geq 0 \quad \text{for all } 1 \leq i \leq n,\ 1 \leq j \leq n.$$

Due to the total unimodularity of the underlying constraint matrix and integrality of the right-hand sides, the vertices satisfy $x_{ij} \in \{0, 1\}$. In fact, they are in one-to-one correspondence to all possible assignments of cars to parking spots. We prove that a linear projection of a mild relaxation of this polytope gives PF_n.

Theorem 8.5.1. *The polytope* PF_n *is a linear projection of a relaxed Birkhoff/assignment polytope.*

Proof. Note that either of the two types of equality constraints in the description of the assignment polytope, combined with the non-negativity constraints, implies an upper bound of $x_{ij} \leq 1$. This allows an equivalent description of the form

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } 1 \leq i \leq n$$

$$\sum_{i=1}^{n} x_{ij} \geq 1 \quad \text{for all } 1 \leq j \leq n$$

$$x_{ij} \geq 0 \quad \text{for all } 1 \leq i \leq n,\ 1 \leq j \leq n.$$

Informally, each car is assigned to precisely one spot, and each parking spot requires at least one car assigned to it, which is only possible if one retains a one-to-one assignment.

We now perform a relaxation where we allow for the assignment of multiple cars to the same (preferred) parking spot, following the design of PF_n: the i^{th} car to arrive picks from the first i spots. To represent the corresponding conditions $\alpha'_i \leq i$ in a nondecreasing rearrangement of the entries, one can replace the set of constraints $\sum_{i=1}^n x_{ij} \geq 1$ by sums of the first k constraints for $1 \leq k \leq n$:

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } 1 \leq i \leq n$$
$$\sum_{j=1}^{k} \sum_{i=1}^{n} x_{ij} \geq k \quad \text{for all } 1 \leq k \leq n$$
$$x_{ij} \geq 0 \quad \text{for all } 1 \leq i \leq n,\ 1 \leq j \leq n.$$

As $\sum_{i=1}^n x_{ij} \geq 1$ for all $1 \leq j \leq n$ implies $\sum_{j=1}^k \sum_{i=1}^n x_{ij} \geq k$ for all $1 \leq k \leq n$, this feasible set, which we call BF_n, is a relaxation of the assignment polytope. Note that the associated constraint matrix lost total unimodularity. As part of our arguments below, we will prove that BF_n is integral, i.e., it has integral vertices.

To obtain PF_n, consider the linear projection of BF_n using $(x_{ij}) \to (\sum_{j=1}^n j \cdot x_{ij})_{i=1,\dots,n}$, i.e., where the i^{th} entry is $\sum_{j=1}^n j \cdot x_{ij}$. For integral (x_{ij}), $x_{ij} = 1$ holds for exactly one j, the parking spot j for car i. It remains to show that the projection of BF_n results in exactly PF_n. Recall that PF_n is an integral/lattice polytope in $\mathbb{R}^n$. Thus, for $\mathbf{a} \in \mathsf{PF}_n$, which may have fractional components, there exists a convex combination $\mathbf{a} = \sum_{\ell=1}^n \lambda_\ell \mathbf{a}^\ell$ of integral parking functions $\mathbf{a}^\ell$. Let $\mathbf{x}^\ell = (x_{ij}^\ell) \in \{0,1\}^{n^2}$ be defined by $x_{ij}^\ell = 1$ if and only if $a_i^\ell = j$. Then $\mathbf{x} = \sum_{\ell=1}^n \lambda_\ell \mathbf{x}^\ell$ lies in BF_n and projects onto $\mathbf{a}$.

Conversely, now let $\mathbf{x} \in \mathsf{BF}_n$. We have to prove that its projection $\mathbf{a}$ lies in PF_n. To this end, we first prove that BF_n is integral. Let $\mathbf{x}$ be fractional and consider a bipartite graph G with $V(G) = C \cup P$, $|C| = n = |P|$, and edges (i,j) from C to P if and only if $0 < x_{ij} < 1$; informally, it is a representation of only the fractional assignments of parking spots to cars. If there exists a cycle in the undirected graph underlying G, one can send flow along either orientation of that cycle and obtain a new feasible solution. In this case, $\mathbf{x}$ is not a vertex of BF_n.

If there does not exist a cycle in G, then there exist at least two nodes with degree 1 in each connected component, as any tree has at least two leaves. By $\sum_{j=1}^{n} x_{ij} = 1$ for all cars i, the degree of $i \in C$ in any component is at least two. Thus, there are at least two parking spots $j', j'' \in P$ with degree 1, as well as a path between j' and j''. Let $j' < j''$ and note that the constraints $\sum_{j=1}^{j'} \sum_{i=1}^{n} x_{ij} \geq j'$ and $\sum_{j=1}^{j''} \sum_{i=1}^{n} x_{ij} \geq j''$ must be satisfied strictly. By sending (maximal) flow along the oriented path from j' to j'', one obtains a new feasible solution with a strictly larger set of active constraints. This again implies that $\mathbf{x}$ is not a vertex of BF_n.

We conclude that BF_n is a lattice/integral polytope with all vertices in $\{0,1\}^{n^2}$. By the same arguments as above, $\mathbf{x} = \sum_{\ell=1}^{t} \lambda_\ell \mathbf{x}^\ell$ is a convex combination with $\mathbf{x}^\ell \in \{0,1\}^{n^2}$. Let $\mathbf{a}^\ell$ be the projection of $\mathbf{x}^\ell$ and observe it is an integral parking function. Thus, $\mathbf{x}$ projects to a convex combination $\mathbf{a} = \sum_{\ell=1}^{n} \lambda_\ell \mathbf{a}^\ell$ of integral parking functions $\mathbf{a}^\ell$. This proves the claim. $\square$

8.5.2 Relation of $\mathfrak{X}_n(\mathbf{b})$ to bounded-size partition polytopes

Most of the proof for Theorem 8.5.1 transfers to $\mathbf{b}$-parking-function polytopes $\mathfrak{X}_n(\mathbf{b})$. By using $x_{ij} \in \{0,1\}$ to indicate car i being assigned a number $1 \leq j \leq S_n$, one can formulate the set of $\mathbf{b}$-parking functions in the form

$$\begin{aligned}
\sum_{j=1}^{S_n} x_{ij} &= 1 \quad \text{for all } 1 \leq i \leq n\\
\sum_{j=1}^{S_k} \sum_{i=1}^{n} x_{ij} &\geq k \quad \text{for all } 1 \leq k \leq n\\
x_{ij} &\geq 0 \quad \text{for all } 1 \leq i \leq n,\ 1 \leq j \leq S_n.
\end{aligned}$$

and the projection $(x_{ij}) \to (\sum_{j=1}^{n} j \cdot x_{ij})_{i=1,\ldots,n}$ gives $\mathfrak{X}_n(\mathbf{b})$ by the same arguments.

A relation of this system to well-known polytopes is more involved than for PF_n. The system can be seen as a relaxation of so-called *bounded-size partition polytopes* [14, 15]. This category of partition polytopes can be considered as special transportation polytopes and generalized Birkhoff or assignment polytopes, representing the (possibly fractional) assignment of n items to ℓ clusters, where one specifies the total weight κ_j assigned to each cluster j. A bounded-size partition polytope specifies lower and/or upper bounds on this weight instead of an exact weight. For the purpose of relating $\mathfrak{X}_n(\mathbf{b})$ to these polytopes, we set $\ell = S_n$ and only use lower bounds. Formally,

$$\begin{aligned} \sum_{j=1}^{S_n} x_{ij} &= 1 \quad \text{for all } 1 \le i \le n \\ \sum_{i=1}^{n} x_{ij} &\ge \kappa_j \quad \text{for all } 1 \le j \le S_n \\ x_{ij} &\ge 0 \quad \text{for all } 1 \le i \le n,\ 1 \le j \le S_n. \end{aligned}$$

The first type of constraints guarantees that each item is assigned to a cluster (or partially to multiple clusters). The second type of constraints guarantees that each cluster receives a certain minimum weight. Note that $S_n > n$ for any $\mathfrak{X}_n(\mathbf{b})$ that is not also a classical parking function PF_n; informally, there are more clusters than items. For the existence of a feasible solution, it is necessary that at least some of the κ_j satisfy $\kappa_j < 1$.

Let $S_0 = 0$ and $t \in [n]$, and set $\kappa_j = \frac{1}{S_t - S_{t-1}}$ for $S_{t-1} < j \le S_t$. One can represent the conditions $\beta_i' \le S_i$ in a nondecreasing rearrangement of the entries by replacing the set of constraints $\sum_{i=1}^{n} x_{ij} \ge \kappa_j$ by sums of the first S_k constraints for $1 \le k \le n$. This gives the above relaxed system.

Corollary 8.5.2. *The polytope $\mathfrak{X}_n(\mathbf{b})$ is a linear projection of a relaxed bounded-size* *partition polytope.*

We conclude by posing the following question to encourage further exploration of generalized parking function polytopes in view of possible relations to other well-known classes of polytopes.

Question 8.5.3. What is the relationship (if any) between $\mathbf{b}$-parking function polytopes and partial permutahedra, polytopes of win vectors, or stochastic sandpile models for graphs?

Part IV

On Arboricity

Chapter Nine

Arboricity Polynomials

The final portion of the dissertation covers the arboricity polynomial, which is a graph invariant. Since graphs have only loosely been mentioned thus far, a brief overview of important graph theory concepts will be included here, as well as the key information on arboricity itself. The work in Section 9.3 and Section 9.4 to advance understanding of arboricity polynomials was accomplished in collaboration with Mark Denker, Alec Helm, Tyrrell McAllister, Sam Spiro, and Andrés R. Vindas-Meléndez during and after the 2023 Graduate Research Workshop in Combinatorics. These results are unpublished.

9.1 Graph Invariants

We will refer to a graph $G = (V, E)$ as having $|V| = n$ vertices and $|E| = m$ edges. A graph invariant is a quantitative representation of some fundamental property of the graph; for example, the degree sequence $(d_1, d_2, ..., d_n), d_1 \geq d_2 \geq ... \geq d_n$ of a graph lists the degrees of the n vertices in non-ascending order. One graph invariant of particular relevance to us is the *chromatic number* $\chi(G)$.

Definition 9.1.1. The **chromatic number** of a graph G, denoted $\chi(G)$, is the smallest number of colors needed to color the vertices of a graph such that no two adjacent vertices have the same color.

We can draw on the language of partitions from Section 2.2 to reinterpret this definition. The "coloring" of the graph vertices is a partition of the n vertices into some number of blocks, such that within any block, none of the vertices are adjacent to each other. This specification is also known as a *proper vertex coloring*. The chromatic number is then the smallest number of blocks that any such partition can have.

A logical follow-up to this is the question of how many such partitions exist for any number of colors k, where k is a positive integer. To answer this question, we introduce an invariant closely related to the chromatic number: the *chromatic polynomial.*

Definition 9.1.2. The **chromatic polynomial** $\chi_G : \mathbb{N} \to \mathbb{N}$ of a graph is a polynomial such that for any $k \in \mathbb{N}$, $\chi_G(k)$ is the number of proper vertex colorings of G using at most k colors.

One of the most useful properties of both the chromatic number and the chromatic polynomial is that they both obey a type of rule called a *deletion-contraction recursion*. There are several such rules for graph invariants. "Deletion" refers to the deletion of some edge $e \in E$ from the graph G, to get a new graph $G \setminus e = (V, E \setminus e)$ on the same set of vertices as G. "Contraction" refers to the merging of the two vertices u, v at the endpoints of some edge e to get a graph G/e; this operation decreases the number of vertices by 1 and also removes e from the edge set.

Definition 9.1.3. A graph invariant $f(G)$ obeys a **deletion-contraction recursion** if for any graph $G = (V, E)$ and any of its edges $e \in E$, $f(G)$ can be written in terms of $f(G/e)$ and $f(G \setminus e)$.

In the specific case of the chromatic polynomial, $\chi_G(k)$ obeys the following recursion for any graph G and any of its edges e:

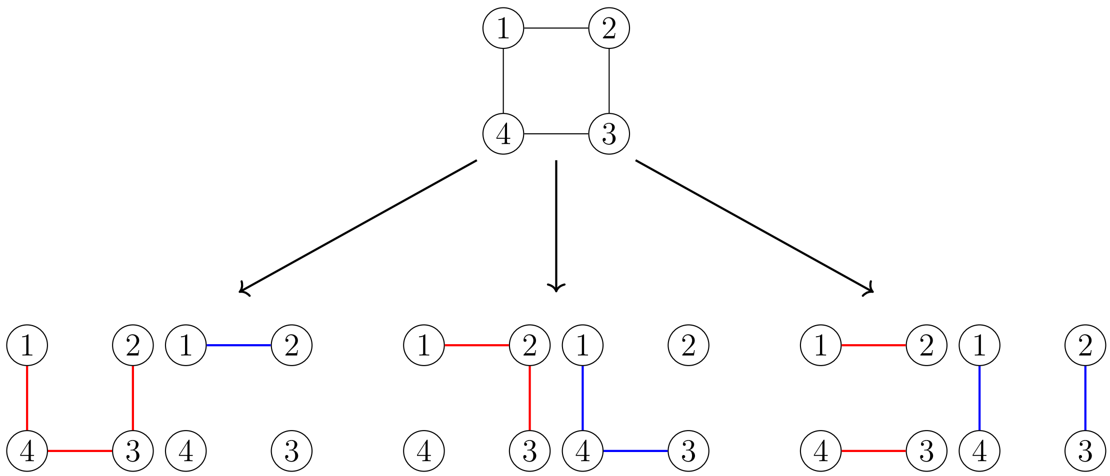
$$\chi_G(k) = \chi_{G\setminus e}(k) - \chi_{G/e}(k)$$

This recursion provides an incredibly powerful tool for computing the chromatic polynomials of large graphs. For any graph, the chromatic polynomial is a function of the polynomials for two slightly smaller graphs, meaning that any chromatic polynomial can be computed by iteratively breaking down the graph into smaller and smaller pieces until graphs with known polynomials are reached. Recursion is similarly powerful for a variety of other well-understood graph invariants; however, there are many interesting invariants that lack such a tool and correspondingly are much harder to break down.

9.2 Arboricity and Scheduling

The arboricity $a(G)$ for a graph G is defined to be the minimum number of forests needed to cover the edges of G. This concept was first introduced as an invariant by Nash-Williams and Tutte [52, 68], and can be naturally extended to any matroid as the minimum number of independent sets needed to cover the ground set. Any tree T will have $a(T) = 1$, as a tree is itself a forest covering its own edges. Any cycle C will have $a(C) = 2$, because the set of edges of a cycle is not a forest, but sequestering even a single edge into a separate second forest will leave a forest behind.

Example 9.2.1. Consider the cycle graph C_4. By definition, the full 4-edge cycle contains a cycle and is therefore not a forest. However, separating the cycle into a 3-edge tree and a 1-edge tree, or into a pair of 2-edge trees, yields two forests that together cover the edges of C_4. Similarly, separating the cycle into a pair of disconnected 2-edge forests will cover the edges of C_4. For visual convenience, within each type of edge partition, the two resulting forests are given distinct colors to separate them in the figure below.



Note above that to achieve a partition of the edges of C_4 into forests denoted by color, in each possible split, every edge is assigned exactly one color. The use of color to identify forests in Example 9.2.1 motivates an alternative understanding of arboricity. We can think of arboricity as behaving like a chromatic number for edges, where a subset of edges can only share a color if it contains no cycles. This interpretation motivates a function $\chi_{A(G)}(k)$ akin to the chromatic polynomial, which is unique for any graph and when evaluated at a positive integer k will give the number of partitions of the edge set into at most k forests, where each forest can be associated with a color. The colorings counted by this function are defined as follows:

Definition 9.2.2. An **arboreal edge coloring** of a graph G is a labeling of the edges of G such that any subset of edges with the same label does not contain a cycle.

We note that these colorings are in general neither proper edge colorings nor acyclic edge colorings. Proper edge colorings require that any two edges incident on the same vertex must have different colors, while acyclic edge colorings require that any two color classes taken together must form a forest. By contrast, arboreal edge colorings only require each individual color to form a forest. A technique for constructing the function $\chi_{A(G)}(k)$ was established by Felix Breuer and Caroline Klivans [45]. The standard chromatic polynomial is proven to be a polynomial using its deletion-contraction recursion, but arboricity does not obey a deletion-contraction recursion, so to prove that $\chi_{A(G)}(k)$ is a polynomial Breuer and Klivans explored its structure via scheduling problems. A scheduling problem consists of a set of m tasks to be completed, subject to some assortment of atomic formulas $x_i \leq x_j, i, j \in [m]$ that give constraints on the order in which the tasks can be done. A k-solution to such a problem consists of an m-dimensional integer point $\mathbf{x}$, whose coordinates possess k distinct values, such that all constraints are satisfied. These integers describe a relative order, rather than an absolute one; if the ordering $x_3 \leq x_1 = x_2$ satisfies the constraints of the problem, then the points $(3, 3, 1)$ and $(15, 15, 7)$ both refer to this single 2-solution. In our context, these problems are useful because they come with well-studied solution-counting polynomials:

Theorem 9.2.3. *Given a scheduling problem S on m items, the scheduling counting function $\chi_S(k)$ is a polynomial in k of degree at most m, taking the following form, where $f_1, ..., f_m$ are non-negative integers counting the number of ordered set partitions with i non-empty blocks satisfying S:*

$$\chi_S(k) = \sum_{i=1}^{m} f_i \binom{k}{i}$$

Scheduling problems can be solved using ordered set partitions (OSPs), which construct relative orderings by setting elements within the same block equal to each other and imposing inequalities between blocks. Any such ordering can be evaluated as a possible solution. Since arboricity is defined via covers made of independent sets, it can also be described using OSPs: any independent cover of a graph's edges is an ordered set partition such that no block of the partition contains a cycle [45].

To capture arboricity for a particular graph G, we can set up a scheduling problem that excludes scenarios where all edges in a cycle occupy a single block, but does not otherwise impose an ordering. The resulting scheduling polynomial evaluates the number of independent covers of the edge set with at most k parts, and is therefore the arboricity polynomial.

Note that in the context of arboricity, the partitions of the edge set do not have any required order. Scheduling problems in general can include conditions that impose orderings on the set partitions, but arboricity does not include any such conditions. Moreover, for any unordered set partition with i non-empty blocks, there are exactly $i!$ corresponding ordered set partitions formed by permuting the order of the blocks. Thus, making use of Theorem 9.2.3, the arboricity polynomial $\chi_{A(G)}(k)$ for some graph G with m edges can be written as $\sum_{i=1}^{m} i! \cdot h_i \binom{k}{i}$, where h_i is the number of unordered partitions of the edge set with i non-empty blocks such that no block contains a cycle. This restructuring of the polynomial given in Theorem 9.2.3 is often more tractable when dealing with arboricity in particular.

Example 9.2.4. Consider again the cycle graph C_4. The only subset of edges which would contain a cycle is the full edge set; therefore, any partition of the edge set with more than 1 block will satisfy the corresponding scheduling problem, while the unique 1-block partition will not. Thus, the arboricity polynomial for C_4 can be computed by counting partitions of $m = 4$. For any set of $m = 4$ objects, there are 7 unordered 2-block partitions, 6 unordered 3-block partitions, and 1 unordered 4-block partitions (refer back to Figure 2.3 for a quick view of the partitions). Making use of these constants, the polynomial will be:

$$\begin{aligned}
\chi_{A(C_4)}(k) &= \sum_{i=2}^{4} i! \cdot h_i \binom{k}{i} \\
&= 2! \cdot h_2 \binom{k}{2} + 3! \cdot h_3 \binom{k}{3} + 4! \cdot h_4 \binom{k}{4}
\end{aligned}$$

$$= 2! \cdot 7 \cdot \frac{k(k-1)}{2!} + 3! \cdot 6 \cdot \frac{k(k-1)(k-2)}{3!} + 4! \cdot 1 \cdot \frac{k(k-1)(k-2)(k-3)}{4!}$$
$$= 7k(k-1) + 6k(k-1)(k-2) + k(k-1)(k-2)(k-3)$$

After some algebra, the expression simplifies to:

$$\chi_{A(C_4)}(k) = k^4 - k$$

It can be shown that in general, the arboricity polynomial for a cycle C_n is $k^n - k = k^m - k$, where $n = m$ is both the number of vertices and the number of edges in the graph. By a similar process, the arboricity polynomial for any tree with m edges is k^m. Both of these results are logical. A tree contains no cycles, and therefore any coloring of its edges will be an arboreal edge coloring, meaning each of the m edges can be colored with any of k colors to get an arboreal edge coloring. By contrast, a cycle graph contains exactly one cycle, which includes all of its edges. If k colors are available, then there are k ways to color all of the edges with the same color, but any other coloring of the k^m total possible colorings will be an arboreal edge coloring. The failure of both arboricity and arboricity polynomials to obey contraction-deletion relationships can be demonstrated from cycle graphs.

Example 9.2.5. Return to C_4 once again. We have established that the arboricity number of any cycle is $a(C_n) = 2$ and that the arboricity polynomial of C_4 is $\chi_{A(C_4)}(k) = k^4 - k$. As shown below in Figure 9.1, contracting a given edge e results in the 3-edge cycle C_3, while deleting an edge results in a 3-edge tree T_3. The contraction and deletion operations result in the following arboricity numbers and polynomials:

$$a(C_4/e) = a(C_3) = 2$$

$$a(C_4 \setminus e) = a(T_3) = 1$$

$$\chi_{A(C_4/e)}(k) = \chi_{A(C_3)}(k) = k^3 - k$$

$$\chi_{A(C_4 \setminus e)}(k) = \chi_{A(T_3)}(k) = k^3$$

Contracting an edge in C_4 produces another cycle, C_3; but all cycle graphs have the same arboricity number, $a(C_n) = 2$, which means there is no way to construct a contraction/deletion relationship that will account for the arboricity number of the deletion graph T_3. Meanwhile, for the polynomial, both deletion and contraction of an edge result in polynomials of order 3, whereas the polynomial of C_4 itself is order 4. This similarly defies the construction of any generalizable formula.

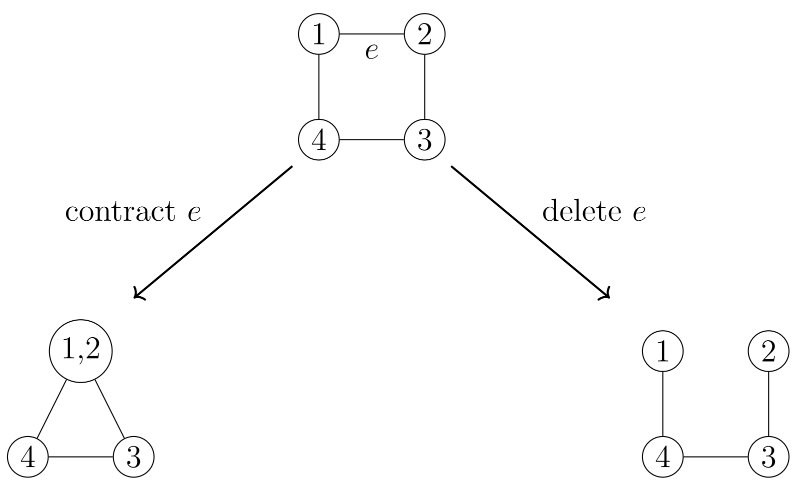


Figure 9.1: Contraction and deletion on C_4

As discussed in Section 9.1, deletion-contraction recursions are incredibly powerful when analyzing and computing graph invariants, and the absence of this tool for arboricity and arboricity polynomials makes them far more arduous to work with.

To alleviate this pressure, an entirely independent collection of analytical rules must be developed to promote understanding of these invariants.

9.3 Results pertaining to graph construction

We begin by observing that the arboricity polynomial of any disjoint union of graphs must be the product of the arboricity polynomials of each individual graph, as any arboreal edge colorings of the disjoint union consist exactly of arboreal edge colorings of each component. This can be extended to a pair of graphs linked by a single edge: the cycles of the larger graph are precisely the sets of cycles of the two smaller graphs, and so arboreal edge colorings are precisely the combinations of arboreal edge colorings of the components along with one of k colors for the connecting edge. This is made rigorous in the following proposition.

Proposition 9.3.1. *Let G be a 1-edge-connected graph, with e being any edge whose deletion disconnects G and G_1, G_2 the components of $G - e$. Then:*

$$\chi_{A(G)}(k) = k \cdot \chi_{A(G_1)}(k) \cdot \chi_{A(G_2)}(k)$$

Proof. Consider a 1-edge-connected graph G and e an edge whose deletion disconnects the graph into connected components G_1, G_2. A cycle is any path, i.e. sequence of adjacent edges, that begins and ends at the same vertex but otherwise has no repeated vertices. In edge terms, this means each edge can only be traversed once in a given cycle. Suppose there existed a cycle in G that contains e. Then after the deletion of e, there would still be a path from one endpoint of e to the other, given by the rest of the cycle. This contradicts the condition that $G - e$ is a disconnected graph. Therefore, removing e does not change the set of cycles in the graph, i.e.

$G - e$ has the same set of cycles as G. The number of arboreal edge colorings of G_1 is given by $\chi_{A(G_1)}(k)$, and the number of arboreal edge colorings of G_2 is given by $\chi_{A(G_2)}(k)$; since $G - e$ consists solely of these two components, any arboreal edge coloring of G_1 can be paired with any arboreal edge coloring of G_2 to get an arboreal edge coloring of $G - e$, which gives:

$$\chi_{A(G-e)}(k) = \chi_{A(G_1)}(k) \cdot \chi_{A(G_2)}(k)$$

For any choice of arboreal edge colorings for G_1 and G_2, the edge e can be given any color; since e does not belong to any cycles of G, its coloring will not alter which edge colorings of G are arboreal. Therefore e can have any of k colors when the arboricity polynomial is evaluated at the integer k, meaning we can simply multiply the polynomial of $G - e$ by k to get the polynomial for G:

$$\chi_{A(G)}(k) = k \cdot \chi_{A(G_1)}(k) \cdot \chi_{A(G_2)}(k)$$

□

In addition to considering 1-edge-connected graphs as whole graphs, we can also examine relationships between the structures of the polynomials for the original graph and the disconnected graph formed by strategic edge deletion. This proof constructively counts ordered set partitions with i blocks for a graph G and its disconnected counterpart, using the fact that for a given number of blocks, adding an edge that cannot create any new cycles yields a predictable number of new ordered set partitions with the same number of blocks.

Proposition 9.3.2. *Let G be a 1-edge-connected graph with m edges, and let e be any edge whose deletion disconnects G. Let $\chi_{A(G)} = \sum_{i=1}^{m} f_i \binom{k}{i}$ be the arboricity polynomial for G and $\chi_{A(G-e)} = \sum_{i=1}^{m-1} f_i^* \binom{k}{i}$ be the arboricity polynomial for $G - e$.*

Then defining $f_0^* := 1$*:*

$$f_i = if_i^* + if_{i-1}^*$$

Proof. Recall from Theorem 9.2.3 that f_i is the number of ordered set partitions of the edge set of G with i non-empty blocks such that no block of any partition contains a cycle. Similarly, the number of OSPs of the edge set of $G - e$ meeting these criteria is given by f_i^*. Note that for any ordered set partition of the edges of G, e has to be somewhere in the partition; either it shares a block with other edges, or it is in a block by itself.

We can observe that for any ordered set partition $\mathcal{P}$ of the edges of G such that e shares a block with other edges, removing e from the partition will yield an ordered set partition of the edges of $G - e$ with the same number of blocks as $\mathcal{P}$. Let $\mathcal{P}_i^*$ be an ordered set partition of $G - e$ with i non-empty blocks. It is known that e is not in any cycles in G, meaning it can be added to any block of $\mathcal{P}_i^*$ to yield an ordered set partition of the edges of G. Since there are i possible blocks in which e could be placed, and this procedure can be undertaken with any such $\mathcal{P}_i^*$, there must be exactly if_i^* ways to construct an ordered set partition of the edges of G with i non-empty blocks from some $\mathcal{P}_i^*$.

Meanwhile, for any ordered set partition $\mathcal{P}$ of the edges of G such that e is in a block by itself, removing e from the partition will yield an ordered set partition of the edges of $G - e$ with one block fewer than $\mathcal{P}$. Let $\mathcal{P}_{i-1}^*$ be an ordered set partition of $G - e$ with $i - 1$ non-empty blocks. Inserting a block containing only e into $\mathcal{P}_{i-1}^*$ will create an ordered set partition of G with i non-empty blocks, because the new block contains only the single edge e and thus does not contain a cycle. The number of ways to insert such a block into $\mathcal{P}^*_{i-1}$ is exactly i, since the block can be added at either end of the ordering or at any of the $i-2$ locations between blocks. This process can be applied to any such partition $\mathcal{P}^*_{i-1}$, meaning there must be exactly if^*_{i-1} ways to construct an ordered set partition of the edges of G with i non-empty blocks from some $\mathcal{P}^*_{i-1}$.

We have established that the ordered set partitions of the edges of G with i non-empty blocks can be separated into two categories – those where e is alone and those where e is in a larger block – and that each of these categories can be counted using the ordered set partitions of the edges of $G-e$. From this, we can conclude that the total number of ordered set partitions of the edges of G with exactly i non-empty blocks is $f_i = if^*_i + if^*_{i-1}$.

□

Proposition 9.3.1 and Proposition 9.3.2 demonstrate one way to analyze arboricity polynomials, which is to identify and characterize edges with special behaviors. A related approach to these polynomials focuses on vertices and their degrees. Any vertex of degree 1, for example, is a leaf, and the single edge incident on it cannot be part of any cycle in the graph. Therefore an arboreal edge coloring of that graph could assign any of the k available colors to such an edge, meaning the arboricity polynomial will have a factor of k for each leaf. This can also be viewed as a consequence of Proposition 9.3.1, since removing the edge to any leaf vertex disconnects that vertex from the rest of its graph. Building on this notion, the next proposition allows vertices of degree 2 to be suppressed. The strategy of pruning a graph to simplify its structure is one that needs further exploration – by finding ways to reduce the complexity of a graph, we can progress towards a toolbox that can analyze small pieces of graphs and assemble the results to generate a fuller understanding of the larger object.

Proposition 9.3.3. *Let G' denote G with an edge e subdivided, then*

$$\chi_{A(G')}(k) = k(k-1)\chi_{A(G-e)}(k) + \chi_{A(G)}(k).$$

Proof. This proof focuses on the ways to obtain arboreal edge colorings of G' based on the information from G. Let e_1 and e_2 be the edges of G' resulting from the subdivision of e in G, and note that they are exactly the two edges incident on the degree-2 vertex v created by the subdivision. In any arboreal edge coloring of G', these two edges either have the same color or different colors.

Suppose e_1, e_2 have the same color in some k-color arboreal edge coloring $\mathcal{C}$ of G'. Consider the edge coloring of G such that e has the same coloring as e_1, e_2 and all other edges are colored the same as in G'. Then because coloring e_1, e_2 with the same color in G' did not yield a cycle of all one color, coloring e with that color must not yield a cycle of all one color in G. Therefore, any such coloring $\mathcal{C}$ of G' must correspond to exactly one arboreal edge coloring of G, of which we know there are $\chi_{A(G)}(k)$ in total.

Suppose e_1, e_2 have different colors in some k-color arboreal edge coloring $\mathcal{C}$ of G'. Note that any cycle containing e in G must instead contain both e_1 and e_2 in G'. By stipulating that e_1 and e_2 have different colors, we ensure that any such cycle will not all be colored the same, even if all other edges in the cycle have the same color as one of the colors used for e_1 and e_2. This is equivalent to removing e from consideration in the coloring altogether, i.e. deleting it from G. Within this framework, there are

$k(k-1)$ ways to color e_1 and e_2, meaning the total number of colorings $\mathcal{C}$ of this type for G' must be $k(k-1)\chi_{A(G-e)}(k)$. Since these cases cover all possible scenarios for a k-color arboreal edge coloring of G', the total number of these colorings must be the sum of the cases:

$$\chi_{A(G')}(k) = k(k-1)\chi_{A(G-e)}(k) + \chi_{A(G)}(k)$$

$\square$

We can return to the cycle graph to see this subdivision property in action. Note that the cycle graph on $m+1$ edges is obtained by subdividing a single edge of C_m, and that subtracting any edge from C_m will yield a path graph on $m-1$ edges, which by definition is a tree. Thus the proposition gives us:

$$\begin{aligned}
\chi_{A(C_{m+1})}(k) &= k(k-1)\chi_{A(T_{m-1})}(k) + \chi_{A(C_m)}(k) \\
&= k(k-1)\cdot(k^{m-1}) + (k^m - k) \\
&= k^{m+1} - k^m + k^m - k \\
&= k^{m+1} - k
\end{aligned}$$

Which is exactly what we expect the polynomial for C_{m+1} to be based on prior results.

Although arboricity does not possess a deletion-contraction recursion, we can still make use of contraction and deletion to develop some intuition about a larger polynomial. Both contraction and deletion are operations that decrease the number of edges in a graph by 1, and since the maximum number of colors that could be used for this missing edge is k, we can multiply the arboricity polynomials of G/e and $G-e$ by a factor of k and compare the resulting polynomials to the arboricity polynomial of G. In fact, these computations bound the arboricity polynomial of G.

Proposition 9.3.4. *For any graph G and any edge $e \in E(G)$, the following inequalities hold for the contraction graph G/e and the deletion graph $G - e$:*

$$k\chi_{A(G/e)}(k) \leq \chi_{A(G)}(k) \leq k\chi_{A(G-e)}(k)$$

Proof. For the first inequality, consider any k-color arboreal edge coloring $\mathcal{C}$ of G/e. Since contraction preserves cycles, G/e has exactly the same number of cycles as G, and any arboreal edge coloring must have at least two colors covering any cycle. Consider the edge coloring of G such that every edge except e has the same color as in $\mathcal{C}$, and e is assigned any of k colors. Then this must be an arboreal edge coloring of G, because it is impossible to assign a color to e and create a monochromatic cycle when all cycles of G/e have already been colored with two or more colors. Therefore G has at least $k\chi_{A(G/e)}(k)$ k-color arboreal edge colorings, i.e. $k\chi_{A(G/e)}(k) \leq \chi_{A(G)}(k)$.

For the second inequality, consider any k-color arboreal edge coloring $\mathcal{C}$ of G. Then any cycle, including those containing e, must be covered by at least two colors. Consider the edge coloring of $G - e$ such that every edge is given the same color as in $\mathcal{C}$; since any cycle in $G - e$ must exist in G, this will be a k-color arboreal edge coloring of $G - e$. Because e has been deleted, it can be restored with any of k colors; however, the result may not be an arboreal edge coloring of G, since one or more of these colors could yield a monochromatic cycle in G. Therefore, G must have at most $k\chi_{A(G-e)}(k)$ k-color arboreal edge colorings, i.e. $\chi_{A(G)}(k) \leq k\chi_{A(G-e)}(k)$. □

From Proposition 9.3.4 we can see that contraction and deletion are still influential and meaningful in the arboricity setting, although they do not have the same power as with other graph invariants that obey strict deletion-contraction recursions. This suggests that although such a formula may be unachievable for arboricity polynomials, there may be a formula that utilizes contraction and deletion in combination with other special behaviors, such as the paradigm in Proposition 9.3.3 that allows vertices with sufficiently small degrees to be suppressed.

9.4 Polynomial properties

To investigate properties of arboricity polynomials and seek out closed formulas for specific families of graphs, we begin by giving a computational exploration of several small graphs whose arboricity polynomials have been explicitly calculated using an algorithm in Python. A table of examples computed in this way is given below. Evaluation for graphs with more edges than those listed here proved difficult due to rapidly-inflating code runtime.

G	$\chi_{A(G)}(k)$
$W_4 = K_4$	$k^6 - 4k^4 - 3k^3 + 12k^2 - 6k$
W_5	$k^8 - 4k^6 - 5k^5 + 6k^4 + 28k^3 - 40k^2 + 14k$
W_6	$k^{10} - 5k^8 - 5k^7 + 9k^6 + 20k^5 + 20k^4 - 120k^3 + 110k^2 - 30k$
W_7	$k^{12} - 6k^{10} - 6k^9 + 15k^8 + 29k^7 - 11k^6 - 30k^5 - 180k^4 + 402k^3 - 276k^2 + 62k$
K_5	$k^{10} - 10k^8 - 15k^7 + 63k^6 + 140k^5 - 540k^4 + 535k^3 - 174k^2$
$K_{2,3}$	$k^6 - 3k^3 + 2k$
$K_{2,4}$	$k^8 - 6k^5 + 8k^3 + 3k^2 - 6k$
$K_{2,5}$	$k^{10} - 10k^7 + 20k^5 + 15k^4 - 30k^3 - 20k^2 + 24k$
$K_{2,6}$	$k^{12} - 15k^9 + 40k^7 + 45k^6 - 90k^5 - 120k^4 + 129k^3 + 130k^2 - 120k$
$K_{3,3}$	$k^9 - 9k^6 + 6k^4 + 36k^3 - 54k^2 + 20k$
$K_{3,4}$	$k^{12} - 18k^9 + 12k^7 + 171k^6 - 162k^5 - 388k^4 + 720k^3 - 420k^2 + 84k$
$K_4 - e$	$k^5 - 2k^3 - k^2 + 2k$

Figure 9.2: Arboricity polynomials for small wheel graphs and bipartite graphs, with a maximum of 12 edges. The graph $K_4 - e$ is also included for completeness.

The examples shown in Figure 9.2 demonstrate that on the surface, the arboricity polynomial do not display many consistent properties in the standard basis. In the table, it can be seen that the signs of the coefficients do not follow a clear pattern and that some terms are completely absent. Two constant characteristics are that the polynomials are monic, and that their orders are equivalent to the number of edges in the corresponding graph. Looking at the construction given in Theorem 9.2.3, we can confirm that these two observations will hold for any graph.

Lemma 9.4.1. *For any graph G with m edges, the arboricity polynomial $\chi_{A(G)}(k)$ will be a monic polynomial of order m.*

Proof. For a graph with m edges, there will be exactly $m!$ ordered set partitions with m non-empty blocks, these being the orderings of m blocks each containing exactly one edge. This is the maximal number of non-empty blocks that any OSP on the edges can have. Moreover, looking at the polynomial structure according to Theorem 9.2.3, each binomial expression produces a polynomial whose order is equal to the number of non-empty blocks in the ordered set partitions being counted, meaning m is the largest possible order. The binomial expression for the order m OSPs produces the order m polynomial:

$$\frac{k!}{m!(k-m)!} = \frac{1}{m!}k \cdot (k-1) \cdot \ldots \cdot (k-m+1)$$

The $m!$ coefficient guaranteed by Theorem 9.2.3 cancels the factor of $m!$ in the denominator, yielding a monic polynomial of order m. Thus, every arboricity polynomial will have a monic leading term of order m. □

However, the lack of niceness elsewhere in the polynomial's structure presents a challenge. The ordered set partition construction suggests an immediate alternative by which some niceness may be recovered. Instead of remaining in the standard basis, we propose moving to the falling factorial basis for further evaluation. Several nice properties emerge by making this adjustment. For notation, we define the falling factorial as

$$k^{\underline{n}} := k(k-1)...(k-n+1)$$

for integers $k \geq 0$. Conversion from the standard basis to the falling factorial basis is executed through the relation $k^n = \sum_{i=0}^{n} S(n,i)k^{\underline{i}}$, or likewise through the relation $k^{\underline{n}} = \sum_{i=0}^{n} s(n,i)k^i$, where $s(n,i)$ and $S(n,i)$ are Stirling numbers of the first and second kinds, respectively.

Proposition 9.4.2. *For any graph G, let $\chi_{A(G)}(k)$ be its arboricity polynomial and let $\underline{\chi}_{A(G)}(k)$ be the representation of $\chi_{A(G)}(k)$ in the falling factorial basis, with $\underline{d}_{min}$ and $\underline{d}_{max}$ being the smallest and largest degrees of non-zero terms in the falling factorial form. Then $\underline{\chi}_{A(G)}(k)$ has the following properties:*

(a) $\underline{\chi}_{A(G)}(k)$ is monic and has the same degree as $\chi_{A(G)}(k)$;

(b) $\underline{d}_{max}$ is the number of edges of the graph G;

(c) for any term of degree $\underline{d}$ in $\underline{\chi}_{A(G)}(k)$, the coefficient represents the number of unordered partitions of the edge set of G with exactly d non-empty blocks such that no block contains a cycle;

(d) $\underline{d}_{min}$ is the arboricity number $A(G)$ of the graph G;

(e) $\underline{\chi}_{A(G)}(k)$ has non-negative coefficients, and moreover all terms of degree $\underline{d}_{min} \leq \underline{d} \leq \underline{d}_{max}$ have strictly positive coefficients.

Proof. Part (1) follows from the conversion of a polynomial in the standard basis to a polynomial in the falling factorial basis. It was observed earlier that the standard arboricity polynomial is monic. Let d_{max} be the degree of $\chi_{A(G)}(k)$, so that the highest-degree term of the polynomial is simply $k^{d_{max}}$. Note that for any degree

d, converting a k^d term to the falling factorial basis yields a polynomial of order d; therefore the only term in $\chi_{A(G)}(k)$ that can yield a term with degree equal to the order of $\chi_{A(G)}(k)$ is the highest-degree term $k^{d_{max}}$, and no terms in $\chi_{A(G)}(k)$ can yield a term with degree greater than this value. Additionally, for any degree d, the coefficient of the highest-degree term in the resulting polynomial is $S(d,d) = 1$. Therefore, the conversion of $k^{d_{max}}$ is guaranteed to produce the monic term $k^{\underline{d}_{max}}$ with $\underline{d}_{max} = d_{max}$, and since the $k^{d_{max}}$ term has a coefficient of 1 in $\chi_{A(G)}(k)$, multiplying this coefficient through the converted polynomial means that this highest-degree term maintains a coefficient of 1, meaning the polynomial is monic in the new falling factorial basis.

Part (2) immediately follows from the first result. Let m be the number of edges in G, and consider the formulation of $\chi_{A(G)}(k)$. Note that there will always exist a single valid m-block partition of the edge set – the partition where each edge is placed in a separate block. Thus $h_m = 1$, and the binomial term gives us a nonzero k^m term, which by the form of $\chi_{a(G)}(k)$ is the highest possible degree that any term could take. Having already established that the order of the polynomial is conserved through conversion to the falling factorial basis, it follows that $\underline{d}_{max} = m$.

For Part (3), we must investigate the structure of the standard arboricity polynomial as well as that of the conversion into the falling factorial basis. In the standard basis, the arboricity polynomial is a sum of binomial terms with coefficients corresponding to set partitions, which can be ordered or unordered depending on perspective. We will focus on the formula using unordered set partitions, which is:

$$\chi_{A(G)}(k) = \sum_{i=1}^{m} i! \cdot h_i \binom{k}{i}$$

Note that $i! \cdot \binom{k}{i} = \frac{k!}{(k-i)!} = k(k-1)...(k-i+1) = k^{\underline{i}}$. Thus this formula immediately becomes:

$$\chi_{A(G)}(k) = \sum_{i=1}^{m} h_i \cdot k^{\underline{i}}$$

Recall that h_i was defined to be the number of unordered set partitions of the edge set with exactly i non-empty blocks such that no block contains a cycle. Since this summation will include exactly one term for each possible value of i, i.e. each possible degree of a term in the polynomial, the coefficient of each term will thus have the desired relationship to the unordered partitions of the edge set.

Part (4) emerges when we consider the falling-factorial formula for the arboricity polynomial as described above. By definition, the arboricity number $A(G)$ of a graph is the smallest number such that an arboreal edge coloring exists. In other words, for any $i < A(G)$, we will have $h_i = 0$ since there will be no unordered set partitions that meet the criteria of an arboreal edge coloring. Similarly, we are guaranteed to have $h_i > 0$ at $i = A(G)$. Therefore there will be a nonzero $k^{\underline{A(G)}}$ term in the arboricity polynomial for any graph G.

Finally, Part (5) also emerges from the falling factorial formula expressed in the proof of Part (3). In this formula structure, we can see that the coefficients of the falling factorial polynomial are simply the constants representing the number of unordered set partitions with i blocks, for i from 1 to m. Since these constants count partitions, no h_i will ever be negative. We have already established that $h_{A(G)}$ must be strictly positive, and that $h_m = 1$ (since the polynomial is monic; this could also follow from an argument on partition structure). For any graph G with m edges and arboricity number $A(G)$, let Φ be an arboreal edge coloring with exactly $k = A(G)$ non-empty blocks, since we know there must exist at least one. If $k = m$, then we are done; note that since there are only m edges on the graph, no partition exists with more than m non-empty blocks. If $k < m$, then Φ must have at least one block containing 2 or more edges. We can construct a new arboreal edge coloring Φ' by dividing this block into two blocks and otherwise using the same blocks as Φ; since Φ is an arboreal edge coloring, none of its blocks contain a cycle, which means subdividing a single block will not yield any blocks containing cycles. (In the language of partitions established earlier in the thesis, we would say that Φ' is a *direct refinement* of Φ.) Then Φ' will have $k+1$ blocks and will be an arboreal edge coloring, meaning $h_{k+1} > 0$. We can repeat this process until we reach $k = m$, meaning there exists at least one arboreal edge coloring with i blocks for every $A(G) \leq k \leq m$, and so all terms of the arboricity polynomial with degrees between $A(G) = \underline{d}_{min}$ and $m = \underline{d}_{max}$ will have strictly positive coefficients. $\square$

To see this conversion and its interpretations in action, we return once again to the cycle graph.

Example 9.4.3. Consider the graph C_4, whose arboricity polynomial is $\chi_{A(C_4)}(k) = k^4 - k$. We can apply the known conversion to see the version of the polynomial in the falling factorial basis, $\underline{\chi}_{A(C_4)}(k)$:

$$\begin{aligned}\underline{\chi}_{A(C_4)}(k) &= \left(\sum_{i=0}^{4} S(4,i)k^{\underline{i}}\right) - \left(\sum_{i=0}^{1} S(1,i)k^{\underline{i}}\right)\\ &= \left(0k^{\underline{0}} + 1k^{\underline{1}} + 6k^{\underline{2}} + 7k^{\underline{3}} + 1k^{\underline{4}}\right) - \left(0k^{\underline{0}} + 1k^{\underline{1}}\right)\end{aligned}$$

$$= k^{\underline{4}} + 7k^{\underline{3}} + 6k^{\underline{2}}$$

We can immediately observe that $\underline{\chi}_{A(C_4)}(k)$ is monic with degree 4, which matches the degree of $\chi_{A(G)}(k)$ and is equal to the number of edges in C_4 as specified by parts (a) and (b) of Proposition 9.4.2. Similarly, the smallest-order term in $\underline{\chi}_{A(G)}(k)$ has degree 2, which is indeed the arboricity number of the cycle C_4 as per part (d) of Proposition 9.4.2. Part (e) is also met by the presence of a degree-3 term with a positive coefficient. However, it is part (c) of Proposition 9.4.2 that now allows us to go beyond verifying known information. According to part (c) of Proposition 9.4.2, the unordered partitions of the edges of C_4 whose blocks avoid cycles will consist of 1 partition with 4 non-empty blocks, 6 partitions with 3 non-empty blocks, and 7 partitions with 2 non-empty blocks. We can refer back to our knowledge of the partitions on $n = 4$ in Section 2.2 to rapidly verify that these counts make sense; since the only cycle in C_4 is the cycle consisting of all 4 edges, every partition on $n = 4$ with more than one block will yield an arboreal edge coloring, and the coefficients match the total numbers of 2-block, 3-block, and 4-block partitions on $n = 4$.

In general, for any cycle C_n, the only unordered partition that will not yield an arboreal edge coloring is the 1-block partition containing all n edges. Since the Stirling numbers of the second kind $S(n, i)$ count the number of unordered partitions of n objects into exactly i blocks, the falling factorial arboricity polynomial for the family of cycle graphs C_n will be:

$$\underline{\chi}_{A(C_n)}(k) = \sum_{i=2}^{n} S(n, i) k^{\underline{i}}$$

This summation produces exactly one term for each power of k, with a clearly-interpretable coefficient in the context of the structure of C_n. The result indicates the potential of the falling factorial basis for improving understanding of arboricity – since this perspective on the polynomials is directly and clearly linked to the structure of the corresponding graph, the rules described in Section 9.3 may have nice extensions when re-evaluated in the falling factorial basis, and similarly new rules may become apparent for other graph structures.

9.5 Future Directions in Arboricity

There are many interesting future investigations to be undertaken in this area. One intriguing direction is to expand on the vertex suppression of Proposition 9.3.3 for higher-degree vertices, ultimately seeking a structure that will allow vertices of any degree to be suppressed. In a similar vein, going beyond the 1-edge-connectedness of the graphs discussed in Proposition 9.3.1 and Proposition 9.3.2 may be fruitful for breaking down arboricity polynomials based on component graphs. Other graph operations, such as gluing graphs along an edge or sequence of edges, could also be investigated. The long-term goal of such endeavors would be to develop a toolbox of techniques for understanding the arboricity polynomial of a graph in terms of the polynomials of smaller graphs, effectively replacing the utility of the deletion-contraction recursion with an array of smaller techniques.

In addition, the many nice properties of the polynomial in the falling factorial basis according to Proposition 9.4.2 indicate that further inquiries into generalized formulas for the arboricity polynomials of families of graphs should be pursued in this alternative basis. Construction may be more straightforward in this setting since the coefficients of the falling factorial polynomial have a direct interpretation in terms of the associated graph, rather than being combinations of factors corresponding to the graph. This also means patterns for coefficient values and polynomial structures may emerge more readily, revealing closer ties between graph structure and cycle counts in the process. A good starting point would be to convert the polynomials in Figure 9.2 into the falling factorial basis to look for patterns among wheel graphs and complete bipartite graphs. Most families of graphs are defined according to the number of vertices they possess; as such, it is prudent to start with graph families whose edge counts grow on the same order as their vertex counts, as well as families that grow in highly predictable ways. The family of complete graphs was excluded from the examples in Figure 9.2 due to having $\binom{n}{2}$ edges, but its extreme regularity in structure makes it a good candidate for further investigation.

Finally, a crucial area for further investigation is the possibility of a reciprocity theorem for arboricity polynomials. Chromatic polynomials, when evaluated at negative integers, have a different combinatorial meaning than when evaluated at positive integers. This represents a broader trend for Ehrhart polynomials, of which chromatic polynomials are merely one category. It can be shown that arboricity polynomials are also Ehrhart polynomials, which means they too should possess a distinct combinatorial interpretation when evaluated for negative integers. However, the nature of this interpretation remains elusive. In Figure 9.3, the evaluations at $k = -1$ are listed for each arboricity polynomial computed and listed in Figure 9.2. These examples do not yield a clear pattern, and their resulting integer sequences produce no results in OEIS, leaving their significance unclear.

As an explicit example of the difficulty with $k = -1$ evaluations, consider the cycle graph C_n, whose arboricity polynomial is $k^n - k$. Evaluating this polynomial at $k = -1$ will result in 2 if n is even, or 0 if n is odd. One sequence which matches

G	$\chi_{A(G)}(-1)$
$W_4 = K_4$	18
W_5	-74
W_6	270
W_7	-914
K_5	-1320
$K_{2,3}$	2
$K_{2,4}$	8
$K_{2,5}$	-8
$K_{2,6}$	112
$K_{3,3}$	-114
$K_{3,4}$	-1272
$K_4 - e$	-2

Figure 9.3: Evaluations of each of the arboricity polynomials given in Figure 9.2 at $k = -1$. No clear pattern emerges for either wheel graphs or bipartite graphs.

this pattern is the Euler characteristics of n-dimensional convex polytopes; however, it currently remains unclear what connection may exist between this sequence and cycle graphs, as well as if or how this observation might translate to more complex graphs. More investigation is required to determine the combinatorial significance of these sequences.

Bibliography

[1] Aruzhan Amanbayeva and Danielle Wang, *The convex hull of parking functions of length n*, Enumer. Comb. Appl. **2** (2022), no. 2, Paper No. S2R10, 10.

[2] Federico Ardila, Carolina Benedetti, and Jeffrey Doker, *Matroid polytopes and their volumes*, Discrete Comput. Geom. **43** (2010), no. 4, 841–854.

[3] Spencer Backman, *Partial graph orientations and the Tutte polynomial*, Adv. in Appl. Math. **94** (2018), 103–119.

[4] Spencer Backman and Gaku Liu, *A regular unimodular triangulation of the matroid base polytope*, 2023, arXiv.2309.10229.

[5] Michel Balinski and Andrew Russakoff, *On the assignment polytope*, SIAM Review **16** (1974), no. 4, 516–525.

[6] Francisco Barahona and Ali Mahjoub, *On the cut polytope*, Mathematical Programming **36** (1986), 157–173.

[7] J. Eric Bartels, John Mount, and Dominic J. A. Welsh, *The polytope of win vectors*, Ann. Comb. **1** (1997), no. 1, 1–15.

[8] Margaret M. Bayer, Steffen Borgwardt, Teressa Chambers, Spencer Daugherty, Aleyah Dawkins, Danai Deligeorgaki, Hsin-Chieh Liao, Tyrrell McAllister, Angela

Morrison, Garrett Nelson, and Andrés R. Vindas-Meléndez, *Combinatorics of generalized parking-function polytopes*, 2024.

[9] Roger E. Behrend, Federico Castillo, Anastasia Chavez, Alexander Diaz-Lopez, Laura Escobar, Pamela Harris, and Erik Insko, *Partial permutohedra*, Sém. Lothar. Combin. **89B** (2023), Art. 64, 12.

[10] D. Bertsimas and J.N. Tsitsiklis, *Introduction to linear optimization*, Athena Scientific, 1997.

[11] Steffen Borgwardt, Jesús A. De Loera, and Elisabeth Finhold, *Edges versus circuits: a hierarchy of diameters in polyhedra*, Adv. Geom. **16** (2016), no. 4, 511–530.

[12] Steffen Borgwardt, Elisabeth Finhold, and Raymond Hemmecke, *On the circuit diameter of dual transportation polyhedra*, SIAM J. Discrete Math. **29** (2015), no. 1, 113–121.

[13] ______, *Quadratic diameter bounds for dual network flow polyhedra*, Math. Program. **159** (2016), no. 1-2, 237–251.

[14] Steffen Borgwardt and Charles Viss, *Constructing clustering transformations*, SIAM J. Discrete Math. **35** (2021), no. 1, 152–178.

[15] Steffen Borgwardt and Charles Viss, *Circuit walks in integral polyhedra*, Discrete Optim. **44** (2022), Paper No. 100566.

[16] Tom Braden, June Huh, Jacob P. Matherne, Nicholas Proudfoot, and Botong Wang, *A semi-small decomposition of the Chow ring of a matroid*, Adv. Math. **409** (2022), Paper No. 108646, 49.

[17] Petter Brändén and Liam Solus, *Symmetric decompositions and real-rootedness*, Int. Math. Res. Not. IMRN (2021), no. 10, 7764–7798.

[18] Benjamin Braun, *Unimodality problems in Ehrhart theory*, Recent trends in combinatorics, IMA Vol. Math. Appl., vol. 159, Springer, [Cham], 2016, pp. 687–711.

[19] Sunil Chopra and M.R. Rao, *Facets of the k-partition polytope*, Discrete Applied Mathematics **61** (1995), no. 1, 27–48.

[20] David L. Davies and Donald W. Bouldin, *A cluster separation measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-1** (1979), no. 2, 224–227.

[21] Jesús A. De Loera, Raymond Hemmecke, and Jon Lee, *On augmentation algorithms for linear and integer-linear programming: from Edmonds-Karp to Bland and beyond*, SIAM J. Optim. **25** (2015), no. 4, 2494–2511.

[22] Antoine Deza and Michel Deza, *On the skeleton of the dual cut polytope*, (1994).

[23] Michel Deza, Martin Grötschel, and Monique Laurent, *Complete descriptions of small multicut polytopes*, Applied Geometry and Discrete Mathematics **4** (1991).

[24] ———, *Clique-web facets for multicut polytopes*, Mathematics of Operations Research - MOR **17** (1992), 981–1000.

[25] Michel Marie Deza and Monique Laurent, *Geometry of cuts and metrics*, 1st ed., Springer Publishing Company, Incorporated, 2009.

[26] Jeffrey Samuel Doker, *Geometry of generalized permutohedra*, University of California, Berkeley, 2011, PhD thesis.

[27] Jack Edmonds, *Submodular functions, matroids, and certain polyhedra*, Combinatorial optimization—Eureka, you shrink!, Lecture Notes in Comput. Sci., vol. 2570, Springer, Berlin, 2003, pp. 11–26.

[28] Farbod Ekbatani, Bento Natura, and László A. Végh, *Circuit imbalance measures and linear programming*, London Mathematical Society Lecture Note Series, pp. 64–114, Cambridge University Press, 2022.

[29] Pedro Felzenszwalb, Caroline Klivans, and Alice Paul, *Iterated linear optimization*, 2021.

[30] ______, *Clustering with semidefinite programming and fixed point iteration*, 2022.

[31] Alan Frieze and Mark Jerrum, *Improved approximation algorithms for max k-cut and max bisection*, 11 1997.

[32] Michel X. Goemans and David P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM **42** (1995), no. 6, 1115–1145.

[33] Branko Grünbaum, *Convex polytopes*, second ed., Graduate Texts in Mathematics, vol. 221, Springer-Verlag, New York, 2003, Prepared and with a preface by Volker Kaibel, Victor Klee and Günter M. Ziegler.

[34] Wakabayashi Grötschel, M. and Y., *Facets of the clique partitioning polytope*, Mathematical Programming **47** (1990), 367–387.

[35] Christian Haase, Andreas Paffenholz, Lindsay C. Piechnik, and Francisco Santos, *Existence of unimodular triangulations—positive results*, Mem. Amer. Math. Soc. **270** (2021), no. 1321, v+83.

[36] Mitsuki Hanada, John Lentfer, and Andrés R. Vindas-Meléndez, *Generalized parking function polytopes*, 2023, Ann. Comb. https://doi.org/10.1007/s00026-023-00671-1.

[37] Dylan Heuer and Jessica Striker, *Partial permutation and alternating sign matrix polytopes*, SIAM J. Discrete Math. **36** (2022), no. 4, 2863–2888.

[38] Johan Håstad, *Some optimal inapproximability results*, **48** (2001), no. 4.

[39] L. Hubert and P. Arabie, *Comparing partitions*, Journal of classification **2** (1985), no. 1, 193–218.

[40] Katharina Jochemko and Mohan Ravichandran, *Generalized permutahedra: Minkowski linear functionals and Ehrhart positivity*, Mathematika **68** (2022), no. 1, 217–236.

[41] Sean Kafer, Kanstantsin Pashkovich, and Laura Sanità, *On the circuit diameter of some combinatorial polytopes*, SIAM J. Discrete Math. **33** (2019), no. 1, 1–25.

[42] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell, *Optimal inapproximability results for max-cut and other 2-variable csps?*, SIAM Journal on Computing **37** (2007), no. 1, 319–357.

[43] Edward D. Kim and Francisco Santos, *An update on the Hirsch conjecture*, Jahresbericht der Deutschen Mathematiker-Vereinigung **112** (2010), no. 2, 73–98.

[44] Jon Kleinberg and Éva Tardos, *Algorithm design*, Addison Wesley, 2006.

[45] Caroline Klivans, *Arboricity polynomials*, in preparation.

[46] Monique Laurent, *Cuts, matrix completions and graph rigidity*, Math. Program. **79** (1997), no. 1–3, 255–283.

[47] Monique Laurent and Svatopluk Poljak, *On a positive semidefinite relaxation of the cut polytope*, Linear Algebra and its Applications **223-224** (1995), 439–461, Honoring Miroslav Fiedler and Vlastimil Ptak.

[48] Yann LeCun and Corinna Cortes, *MNIST handwritten digit database*, (2010).

[49] Hsin-Chieh Liao, *Stembrige codes and Chow rings*, Sém. Lothar. Combin. **89B** (2023), Article #88, 12pp.

[50] Fu Liu, *On positivity of Ehrhart polynomials*, Recent trends in algebraic combinatorics, Assoc. Women Math. Ser., vol. 16, Springer, Cham, 2019, pp. 189–237.

[51] Dustin G. Mixon, Soledad Villar, and Rachel Ward, *Clustering subgaussian mixtures by semidefinite programming*, 2016.

[52] C. St.J. A. Nash-Williams, *Decomposition of Finite Graphs Into Forests*, Journal of the London Mathematical Society **s1-39** (1964), no. 1, 12–12.

[53] James G. Oxley, *Matroid Theory (Oxford Graduate Texts in Mathematics)*, second ed., Oxford University Press, 2011.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research **12** (2011), 2825–2830.

[55] Alex Postnikov, Victor Reiner, and Lauren Williams, *Faces of generalized permutohedra*, Doc. Math. **13** (2008), 207–273.

[56] Alexander Postnikov, *Permutohedra, associahedra, and beyond*, Int. Math. Res. Not. IMRN (2009), no. 6, 1026–1106.

[57] R. Tyrrell Rockafellar, *The elementary vectors of a subspace of* R^N, Combinatorial Mathematics and its Applications, University of North Carolina Press, Chapel Hill, NC, 1969, pp. 104–127.

[58] Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor, *Adjusting for chance clustering comparison measures*, 2015.

[59] Alexander Schrijver, *Combinatorial optimization: polyhedra and efficiency*, vol. 24, Springer Science & Business Media, 2003.

[60] Thomas Selig, *The stochastic sandpile model on complete graphs*, 2022, arXiv.2209.07301.

[61] Lloyd S Shapley, *Cores of convex games*, International journal of game theory **1** (1971), no. 1, 11–26.

[62] Richard P. Stanley, *Problem 12191*, Problems and Solutions, Amer. Math. Monthly **127** (2020), no. 6, 563–571.

[63] Richard P. Stanley and Jim Pitman, *A polytope related to empirical distributions, plane trees, parking functions, and the associahedron*, Discrete Comput. Geom. **27** (2002), no. 4, 603–634.

[64] The Sage Developers, *Sagemath, the Sage Mathematics Software System (Version 10.2)*, 2024, `https://www.sagemath.org`.

[65] Donald M. Topkis, *Adjacency on polymatroids*, Mathematical Programming **30** (1984), no. 2, 229–237.

[66] ———, *Paths on polymatroids*, Math. Programming **54** (1992), no. 3, 335–351.

[67] Csaba D. Toth, Jacob E. Goodman, and Joseph O'Rourke, *Handbook of discrete and computational geometry, third edition*, third edition. ed., Discrete Mathematics and Its Applications, CRC Press, Boca Raton, FL, 2017 (eng).

[68] W. T. Tutte, *On the Problem of Decomposing a Graph into n Connected Factors*, Journal of the London Mathematical Society **s1-36** (1961), no. 1, 221–230.

[69] Nguyen Xuan Vinh, Julien Epps, and James Bailey, *Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance*, J. Mach. Learn. Res. **11** (2010), 2837–2854.

[70] Catherine H. Yan, *On the enumeration of generalized parking functions*, Proceedings of the Thirty-first Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 2000), vol. 147, 2000, pp. 201–209.

[71] ______, *Generalized parking functions, tree inversions, and multicolored graphs*, Adv. in Appl. Math. **27** (2001), no. 2-3, 641–670, Special issue in honor of Dominique Foata's 65th birthday (Philadelphia, PA, 2000).

[72] ______, *Parking functions*, Handbook of enumerative combinatorics, Discrete Math. Appl. (Boca Raton), CRC Press, Boca Raton, FL, 2015, pp. 835–893.

[73] Vladimir A. Yemelichev, M.M. Kovalëv, and M.K. Kravtsov, *Polytopes, graphs and optimisation*, Cambridge University Press, 1984, Translated from the Russian by G.H. Lawden.

[74] Günter M. Ziegler, *Lectures on polytopes*, Graduate Texts in Mathematics, vol. 152, Springer-Verlag, New York, 1995.